



**Universidade de
Aveiro**
2018

Departamento de Comunicação e
Arte

**Nuno Miguel Almeida
Soares**

**Desenvolvimento de uma aplicação
móvel de suporte ao Programa de
Tutoria da Universidade de Aveiro**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Comunicação Multimédia, realizada sob a orientação científica da Doutora Dora Maria de Oliveira Simões Ribeiro Pereira, Professora Adjunta do Instituto Superior de Contabilidade e Administração da Universidade de Aveiro e coorientação da Doutora Gillian Grace Owen Moreira, Professora Auxiliar do Departamento de Línguas e Culturas da Universidade de Aveiro.

Dedico este trabalho a todos os meus professores deste Mestrado, e a todos os professores que ao longo de todos estes anos me transferiram conhecimentos nas mais diversas áreas. Dedico este trabalho à minha família, amigos, e às minhas orientadoras nesta Dissertação de Mestrado.

o júri

presidente

Prof. Doutora Lúdia de Jesus Oliveira Loureiro da Silva,
Professora Associada C/ Agregação, Universidade de Aveiro (Presidente)

Prof. Doutor Leonel Caseiro Morgado,
Professor Auxiliar Convidado Com Agregação, Universidade Aberta (Vogal)

Prof. Doutora Dora Maria de Oliveira Simões Ribeiro Pereira,
Professora Adjunta, Universidade de Aveiro (Vogal)

agradecimentos

Este trabalho não seria possível sem o contributo e ajuda das orientadoras, que se mostraram sempre disponíveis para debater os mais diversos tópicos relacionados com a dissertação.

palavras-chave Aplicações móveis, Programa de Tutoria, Laravel, React, Redux, React Native.

resumo Atualmente, assiste-se a uma grande utilização dos dispositivos móveis, como os *smartphones*, por parte dos jovens, para comunicarem e partilharem experiências em redes sociais, mesmo durante as aulas. O Programa de Tutoria da Universidade de Aveiro (PT-UA), criado no ano letivo de 2010/2011, coloca em prática um conjunto de medidas que visam a integração e acompanhamento adequado dos novos estudantes (tutorandos). Esse acompanhamento é feito através de mentores e tutores.

Face ao cenário descrito, a presente dissertação tem como objetivo o desenvolvimento de uma aplicação móvel (*App*), e da respetiva versão web, de suporte ao PT-UA.

A metodologia de investigação suporta-se na Investigação de Desenvolvimento. O processo de desenvolvimento engloba o estado da arte sobre a utilização de websites de redes sociais e aplicações móveis na educação, e principais funcionalidades de software; a conceção dos *layouts*; a avaliação através de testes de usabilidade e a implementação das sugestões de melhoria.

A solução final apresentada permite uma melhor comunicação entre todos os intervenientes do programa, através da sua acessibilidade em qualquer lugar, de forma rápida. Apresenta uma rede social, onde todos os participantes podem publicar e partilhar ideias através de *posts*, que podem ser comentados. Oferece navegação GPS para facilitar a movimentação pelo campus. Possibilita o acesso ao calendário de eventos, assim como a possibilidade de *chat* em tempo real. São utilizadas várias tecnologias modernas, nomeadamente o React Native e Redux para a *App*, e o Laravel e Pusher para a parte da versão *web* e *chat* em tempo real. Relativamente às bases de dados optou-se pela utilização do MySQL, por existir uma maior familiaridade.

keywords Mobile Application, Tutoring Program, Laravel, React, React Native, Redux.

abstract Today, mobile devices such as smartphones are widely used by young people to communicate and share experiences on social networks, even during classes. The University of Aveiro (UA) created the Tutoring Program of the University of Aveiro (PT-UA) in the academic year 2010/2011, putting in practice a set of measures that aim at the integration and adequate support of new students during their first year of studies. This support is provided by tutors and mentors.

This dissertation aims to develop a mobile application (App) and corresponding web version to support the PT-UA.

It uses the methodology of Research Development. The development process includes a state of the art of the use of social network websites and mobile applications in education, and principal software functionalities; layout design; usability testing and the implementation of suggestions for improvement.

The solution presented allows better communication between all the actors of the program, by enabling rapid accessibility anywhere, only requiring a smartphone and internet access. It features a social network, where everyone can post and share ideas via posts, which can be commented on and liked by other users. It provides GPS navigation to make it easier for users to navigate the campi, and it gives access to a calendar of events, as well as the possibility to chat in real time with other participants in the program. Several modern technologies were used, react native and redux for the App, and laravel and pusher for the back-office and real-time chat. Regarding the databases, it was decided to use MySQL, because of greater familiarity.

Índice de Conteúdos

Índice de Conteúdos	ii
Índice de Tabelas	iv
Índice de Figuras	vi
Índice de Abreviaturas.....	viii
1. Introdução.....	1
2. Enquadramento teórico.....	5
2.1 Era digital no contexto dos programas de tutoria	5
2.1.1 Desafios conjuntais e sociais.....	5
2.1.2 Evolução dos serviços e plataformas de utilização.....	8
2.1.3 Linguagens e software de desenvolvimento	9
2.2 Aplicação móvel	16
2.2.1 Definição e princípios de desenvolvimento	17
2.2.2 Ferramentas para <i>mockup</i>	24
2.2.3 Formas de desenvolvimento.....	25
2.3 Plataformas e aplicações de suporte a programas de tutoria.....	28
2.4 Considerações finais	34
3. Metodologia.....	37
3.1 Objetivos e abordagem de investigação.....	37
3.2 Conceção dos layouts.....	39
3.3 Recolha e análise de dados	40
4. Resultados.....	47
4.1 Contributo da App PT-UA.....	47
4.2 Implementação	49
4.2.1 Versão <i>App</i> smartphone	49
4.2.2 Versão plataforma <i>web</i>	58
4.3 Testes de usabilidade	60

4.4 Melhorias implementadas e a implementar	68
5. Conclusões finais.....	71
Bibliografia.....	75
Apêndice A: Disponibilização do código.....	85

Índice de Tabelas

Tabela 1 – Funcionalidades das principais redes sociais utilizadas em Portugal.....	9
Tabela 2 - Características de <i>software</i> para <i>mockup</i>	25
Tabela 3- Comparação de arquiteturas de <i>software</i> REST e SOAP.....	27
Tabela 4 - Comparação de várias plataformas de apoio a programas de tutoria.....	35
Tabela 5 – Guião fornecido aos participantes do estudo.....	43
Tabela 6 – Questões colocadas aos participantes do teste de usabilidade.....	44
Tabela 7 – Distribuição dos participantes por cargo no PT-UA.....	45
Tabela 8 – Resumo das sugestões de melhoria.....	69

Índice de Figuras

Figura 1 – Primeira página <i>Web</i> , apenas com HTML (Arah, 2005).	10
Figura 2 – Exemplo de uma página <i>Web</i> com CSS (ieatcss.com, n.d.).....	10
Figura 3 – Biblioteca <i>Javascript</i> que adiciona efeitos visuais (Jqueryhouse.com, n.d.).....	11
Figura 4 – Diferenças de sintaxe do ES5 para o ES6 (Shappir, 2016).	12
Figura 5 – Exemplo da utilização do PHP pela linha de comandos (Arda, 2014).	13
Figura 6 - Alguns ecrãs da plataforma Unibly (Unibly, n.d.).....	31
Figura 7 - Alguns ecrãs da plataforma Mentored (Mentored.com, n.d.).....	32
Figura 8 - Alguns ecrãs da plataforma Edmodo (Edmodo, 2018).....	33
Figura 9 - Alguns ecrãs da aplicação da Universidade de Ashford (Education, n.d.).....	33
Figura 10 - Fases de investigação.....	38
Figura 11 – Distribuição dos participantes por Unidade Orgânica.	44
Figura 12 – Principais valores do PT-UA (Programa de Tutoria / PT-UA, 2018).....	48
Figura 13 – <i>Landing screen</i> da <i>App</i> PT-UA e ecrã de registo, respetivamente.....	50
Figura 14 - Seleção do cargo de utilizador e o feedback após o registo, respetivamente.	50
Figura 15 – <i>Feedback</i> visual durante o processo de <i>login</i> no ecrã Entrar.	51
Figura 16 – Ecrã Notícias.	51
Figura 17 – Inserção de um comentário a um dos <i>posts</i> do ecrã Comunidade.....	52
Figura 18 – Inserção de uma mensagem de texto no ecrã de <i>Chat</i>	53
Figura 19 – <i>Screenshots</i> do ecrã Mapa e suas principais funcionalidades.	53
Figura 20 – <i>Screenshots</i> do ecrã de Perfil e a alteração dos dados do utilizador.	54
Figura 21 – Ecrã de Eventos com a listagem e a criação de um evento, respetivamente.....	54
Figura 22 – Esquema do funcionamento do <i>Redux</i> na <i>App</i> PT-UA.	56
Figura 23 – Esquema elucidativo navegação do utilizador na <i>App</i> PT-UA.	57
Figura 24 – Ecrã Notícias da versão <i>Web</i> , destinada ao uso em Desktop.....	58
Figura 25 – Ecrã de <i>Chat</i> da versão <i>Web</i>	59
Figura 26 – Ecrã de Eventos com o formulário de criação de eventos.	59

Figura 27 – Ecrã de Perfil da versão <i>Web</i>	60
Figura 28 – Alteração dos dados de utilizador no ecrã de Perfil, versão <i>Web</i>	60
Figura 29 – Grau de satisfação geral na utilização da <i>App</i>	61
Figura 30 – Grau de satisfação referente à navegação inferior e lateral.	62
Figura 31 – Grau de dificuldade relativamente à navegação (inferior e lateral).	62
Figura 32 – Grau de dificuldade relativamente à criação de uma nova conta.	62
Figura 33 – Grau de dificuldade relativamente à criação de uma nova notícia.	63
Figura 34 – Grau de satisfação do feed de notícias e suas funcionalidades.	63
Figura 35 – Grau de satisfação relativamente ao ecrã de perfil e edição de dados.	64
Figura 36 – Grau de dificuldade na edição dos dados de perfil.	64
Figura 37 – Grau de satisfação relativamente ao ecrã de <i>chat</i>	64
Figura 38 – Grau de dificuldade da utilização do <i>Chat</i>	65
Figura 39 – Grau de satisfação do ecrã de Mapa Navegação.	65
Figura 40 – Grau de dificuldade relativamente do mapa de Navegação.	66
Figura 41 – Grau de dificuldade relativamente à criação de um novo evento.	66
Figura 42 – Grau de satisfação relativamente ao ecrã de eventos e criação de novos conteúdos. ...	67

Índice de Abreviaturas

3E – *Expressing Experiences and Emotions*

AXE – *Anticipated eXperience Evaluation*

CDN – Content Delivery Network

CSRF – *Cross-Site Request Forgery*

CSS – Cascading Style Sheets

DBio – *Departamento de Biologia da Universidade de Aveiro*

DETI – *Departamento de Electrónica, Telecomunicações e Informática*

DFis – *Departamento de Física da Universidade de Aveiro*

DGES – Direção Geral do Ensino Superior

DNS – *Domain Name System*

DRY – *Don't Repeat Yourself*

ESTGA – *Escola Superior de Tecnologia e Gestão de Águeda*

HDD – *Hard Disk Drive*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

IES – Instituições de Ensino Superior

IP – *Internet Protocol*

ISCTE – Instituto Universitário de Lisboa

MAX – *Method of Assessment of eXperience*

ORM – *Object Relational Mapping*

PE – Programação Estruturada

PHP – *Hypertext Preprocessor*, originalmente *Personal Home Page*

POA – Programação Orientada a Aspetos

POO – Programação Orientada a Objetos

RAID – *Redundant Array of Inexpensive Drives*

SAM – *Self Assessment Manikin*

SDD – *Solid State Drive*

TDD – *Test Driven Development*

UA – Universidade de Aveiro

UC – Unidade Curricular

UI – *User Interface*

URL – *Uniform Resource Locator*

UX – *User eXperience*

XSS – *Cross-Site Scripting*

1. Introdução

Em 1999 surgiu a Declaração de Bolonha com o intuito de criar um Espaço Europeu de Ensino Superior, coerente e competitivo através de um sistema de graus compatíveis, utilização de um sistema de créditos e promoção da mobilidade (Pływaczewski & Kraśnicka, 2016). Em Portugal, só em 2007 esta declaração começou a ser aplicada nos planos curriculares (Universidade de Coimbra, 2010). Resultante da sua aplicação, verificou-se, entre outras coisas, uma maior preocupação para compreender e combater as causas do abandono escolar, através de um acompanhamento contínuo dos novos estudantes nos sistemas de ensino. No relatório anual do ATTRACT é referido que o abandono escolar é influenciado pela interação entre os docentes, estudantes e os estabelecimentos de ensino (ATTRACT Project, 2012). Existe uma necessidade de se fazer um acompanhamento dos estudantes, principalmente no primeiro ano, onde a taxa de abandono é mais acentuada. Durante o primeiro ano, os estudantes aprendem novas competências e normas, sendo que esta nova aprendizagem pode ser facilitada com o contributo de estudantes de anos posteriores, que podem partilhar a sua experiência.

A Universidade de Aveiro (UA) tem um historial de iniciativas relacionado com a Tutoria que data desde o ano letivo de 1997/98. A primeira iniciativa contemplava as figuras do Professor Conselheiro e Estudante Conselheiro, ocorrendo um acompanhamento próximo, procurando reduzir a taxa de insucesso e abandono escolar. Em 2001 foi introduzido o conceito de Orientador de Estudos, onde o orientador auxiliava os estudantes nos seus estudos e organização da vida académica, facilitando a integração (Fernandes, 2013).

O Programa de Tutoria da Universidade de Aveiro (PT-UA)¹ surge no ano letivo de 2010/2011, colocando em prática um conjunto de medidas que visam a integração e o acompanhamento adequado dos novos estudantes (tutorandos), com vista à promoção de uma cultura de sucesso, de bem-estar e de aprendizagem de competências transversais (Simões, Pinheiro, & Moreira, 2016). O PT-UA assegura o acompanhamento do estudante, auxiliando-o na transição do ensino secundário para o superior. Esse acompanhamento é feito através de mentores (estudantes mais velhos), supervisionados por tutores (docentes), facilitando o processo de integração na comunidade académica. Para a consolidação do sucesso deste programa entende-se que a promoção de meios de comunicação, que facilitem o diálogo entre as várias partes, é fundamental para que as informações possam chegar atempadamente e adequadamente aos principais interessados (Fernandes, 2013; Simões et al., 2016).

¹ <http://www.ua.pt/tutoria/page/18959>

Por seu lado, o aparecimento da Internet, o seu rápido crescimento e o nascimento dos *smartphones* e outros dispositivos móveis têm vindo a revolucionar a forma como percebemos o mundo e nos comportamos em sociedade. Perante um cenário de grandes alterações tecnológicas e sociais, torna-se pertinente analisar os seus impactos no sistema de ensino. Atualmente, uma quantidade considerável de estudantes utiliza aparelhos de conexão à Internet durante as aulas para navegar e consultar as últimas novidades em redes sociais, como o *Facebook* e o *Instagram* (Kadry & Roufayel, 2017). A grande adoção das redes sociais *online* tem levado a que os professores procurem integrar estas novas tecnologias digitais nas suas aulas. Alguns estudos referem que os estudantes gostam de ter *feedback online* e da possibilidade de terem suporte imediato (p.e., Pimmer et al., 2017), referindo nomeadamente o uso do WhatsApp para poderem ouvir e ao mesmo tempo terem direito a expressar as suas opiniões, facilitando a aprendizagem e colaboração fora das salas de aulas. A maioria da aprendizagem feita com recurso a redes sociais é constituída pelas interações entre os estudantes e professores. A taxa de participação está relacionada com o grau de confiança que os estudantes têm com os professores, sendo que quando estes interagem pessoalmente com um professor têm tendência em participar mais nas redes sociais *online* (Pimmer et al., 2017).

No contexto de um programa de tutoria, os contactos entre tutores, mentores e tutorandos (estudantes do 1.º ano) deve ser contínuo, tanto físico como à distância. Plataformas de *chat*, websites de redes sociais, ou outras plataformas similares apresentam-se, à partida, como facilitadoras. A utilização de tecnologias de videoconferência pode ser também uma ótima forma de envolvimento (*engagement*) e de potenciar as relações que se estabelecem (Merritt & Havill, 2016). O aumento da importância e utilização das redes sociais e *smartphones* por parte dos estudantes despoleta assim a necessidade de desenvolver uma *App* de apoio ao PT-UA, que potencialize o contacto entre os tutores, mentores e tutorandos nesses meios.

No âmbito de um trabalho de investigação científica é importante diferenciar os objetivos de investigação das orientações metodológicas. Os objetivos de investigação incluem vários fatores responsáveis pela motivação por parte do investigador em investigar o tema. O paradigma em que este recebeu a formação influencia as escolhas metodológicas (Coutinho, 2006). Assim, e num contexto em que o autor desta investigação, embora tendo iniciado o seu percurso académico com uma licenciatura em *Marketing*, tenha, de forma autodidata, aprendido a programar, surge a motivação para o tema de investigação que se apresenta. Os conceitos adquiridos pelo autor durante as várias cadeiras do mestrado, permitiram que este tivesse a confiança e conhecimento necessário a desenvolver novos projetos. Esta alia-se ainda ao facto de o autor ter participado no

Avila Crew², como mentor, e a sua atuação assentar numa preocupação de que os novos estudantes não se sentissem desmotivados e desejar o seu sucesso. Assim, desenhar e implementar uma aplicação que permita suportar o PT-UA apresenta-se como o objetivo principal da presente dissertação. Além de uma versão *Web* que permita facilitar a gestão de conteúdos e utilizadores, pretende-se potencializar o PT-UA com a integração de novas tecnologias e da vertente de redes sociais, aproximando-se mais do seu público-alvo.

Atendendo a que o objetivo principal desta dissertação é o desenho e implementação de uma plataforma de suporte ao PT-UA, apresentam-se como principais objetivos intermédios:

- Análise do estado de arte.
- Identificação das normas, cuidados, *standards* de mercado relativamente ao *design* de aplicações móveis.
- Identificação e seleção das tecnologias, *frameworks* e bases de dados mais adequadas.
- Implementação do protótipo em código, plenamente funcional, e realização de testes de usabilidade.
- Implementação das melhorias sugeridas e identificação de outras que possam ser realizadas em trabalhos de investigação futuros.

O relato do trabalho desenvolvido é feito no presente documento, estruturado em cinco capítulos, sendo este o primeiro capítulo. Neste primeiro capítulo é feita uma breve introdução ao tema e apresentação do PT-UA. É também referida a crescente utilização por parte dos jovens de *smartphones*, e justificada a pertinência e motivação para a investigação.

No segundo capítulo é feito um enquadramento teórico, sendo este constituído por três subcapítulos. No primeiro é realizada uma resenha sobre a era digital no contexto dos programas de tutoria, nomeadamente pelos desafios conjuntais e societais, a evolução dos serviços e plataformas de utilização, linguagens e boas práticas de desenvolvimento de software. No segundo subcapítulo são apresentadas as aplicações móveis, abordando-se os princípios de desenvolvimento de aplicações, algumas ferramentas para o *mockup*, as várias formas de desenvolvimento, cuidados a ter no *design* de interfaces e bases de dados. No terceiro subcapítulo são apresentadas algumas plataformas e aplicações de apoio a programas de tutoria.

O terceiro capítulo é dedicado à metodologia de investigação, onde se relembram os objetivos de investigação e se descrevem os procedimentos de conceção dos *layouts* e de recolha e análise de dados.

² Iniciativa que visa combater o insucesso escolar na Unidade Curricular (UC) de Laboratório Multimédia 4 da Licenciatura em Novas Tecnologias da Comunicação (NTC).

O quarto capítulo é dedicado à apresentação dos resultados, sendo composto por quatro subcapítulos. Primeiramente apresenta-se em maior detalhe o PT-UA e a sua forma de funcionamento. Após o que se descrevem os principais aspetos da implementação da *App* PT-UA e da versão *website*. Posteriormente descrevem-se os resultados obtidos nos testes de usabilidade. E finalmente, apresentam-se as principais sugestões de melhoria identificadas.

No quinto capítulo apresentam-se as conclusões finais, sendo realçadas algumas limitações do estudo e os principais contributos da *App* para o PT-UA. Indicam-se ainda possíveis direções de investigação futura.

2. Enquadramento teórico

Neste capítulo é feito o enquadramento teórico através de três subcapítulos onde se apresentam a era digital no contexto dos programas de tutoria, as formas de *mockup* e desenvolvimento de aplicações, e as plataformas e aplicações de apoio a programas de tutoria já existentes atualmente pelo mundo.

2.1 *Era digital no contexto dos programas de tutoria*

A exposição dos principais desafios conjunturais e sociais da era digital no ensino, e em particular no contexto de programas de tutoria, antecede uma breve resenha da evolução da Web nos últimos anos. Apresentam-se os principais avanços no desenvolvimento de novas linguagens de programação, e que pretendem responder ao aumento das exigências dos utilizadores. Antes de se falar do *design* e de como conceber aplicações móveis, é necessário compreender as tecnologias *web* e como estas evoluíram, uma vez que são base de muitos serviços *web* que as aplicações utilizam para disponibilizar e aceder aos conteúdos. Indicam-se, ainda, os principais desafios que surgem quando se desenvolvem aplicações.

2.1.1 Desafios conjuntais e sociais

As alterações do comportamento dos estudantes decorrentes da era digital trouxeram novos desafios às Instituições de Ensino Superior (IES) e, em particular, à implementação de programas de tutoria. Apresentando-se alguns programas em Portugal de promoção de sucesso académico e de combate ao abandono escolar, apontam-se algumas vantagens da integração das redes sociais e das novas tecnologias a programas já existentes.

Entende-se, neste trabalho, por Programa de Tutoria, todo o projeto que vise a promoção do sucesso académico e o combate ao abandono escolar, através de um acompanhamento dos novos estudantes, realizado por tutores e/ou mentores, assim como a existência de formações que dotem os novos estudantes de competências transversais. Entende-se também como Programa de Tutoria o auxílio por parte de tutores e mentores na integração dos novos estudantes na comunidade académica, facilitando a transição para o Ensino Superior. Esse acompanhamento pode ser realizado de várias formas, sendo importante a existência de confiança entre todos os participantes, para permitir a partilha de conhecimentos com os novos estudantes. No caso da Universidade de Aveiro, o PT-UA articula-se com o projeto FICA (Ferramentas de Identificação e Combate ao Abandono), conjunto de mecanismos que procura monitorizar, compreender, identificar e prevenir o abandono e insucesso escolar.

Antes do desenvolvimento de uma *App* de apoio ao programa, é importante compreender as causas, as dificuldades, e as medidas que são tomadas por estes programas de forma a obter uma melhor compreensão das suas necessidades. Nesse sentido é feito um levantamento de alguns programas em Portugal, sendo apontados alguns dos motivos por detrás do abandono escolar, assim como são apresentados alguns casos de sucesso nesse combate.

Em Portugal, o projeto FICA, financiado pelo Ministério da Educação e Ciência, e implementado na Universidade de Aveiro, luta pela diminuição do abandono e melhoria do sucesso escolar, através de um conjunto de medidas que incluem o acolhimento e acompanhamento dos estudantes, e a monitorização dos seus percursos, bem como a promoção de técnicas de ensino inovadoras (Universidade de Aveiro, 2016). Também o Instituto Politécnico de Setúbal implementa um programa de apoio intensivo aos estudantes do primeiro ano, organiza seminários, oficinas, conferências para o debate das causas do insucesso escolar e para que possam ser tomadas medidas (Instituto Politécnico de Setúbal, 2015). Quando um estudante anula a matrícula, o Sistema de Ação Social contacta-o para tentar reverter a situação. Desenvolve ações de formação pedagógica aos docentes e implementa ações de melhoria nas Unidades Curriculares (UC) com pior desempenho. Estudos realizados na Universidade de Coimbra apontam que estudantes do género masculino, mais velhos do que a média, estudantes deslocados e estudantes de licenciatura são quem maioritariamente abandona os estudos. No entanto, entre os caloiros, há mais raparigas a abandonar o sistema de ensino. Esta universidade implementa também um conjunto de medidas de combate ao abandono escolar e diminuição do risco de prescrição. Nomeadamente, disponibiliza formações para que os estudantes possam adquirir competências essenciais, e dinamiza medidas que permitem um acompanhamento mais próximo dos estudantes com pior desempenho (Alarcão, 2015). O Instituto Universitário de Lisboa (ISCTE) refere também a implementação de várias medidas para o combate ao insucesso escolar, como o trabalho de equipa entre Mentores e Tutores, em regime de voluntariado, e ações de formação, oferta de *workshops* e seminários, além de boa qualidade das instalações (Carvalhosa, n.d.).

As medidas indicadas anteriormente vão ao encontro das orientações da Direção Geral do Ensino Superior (DGES), de que os alicerces para a promoção do sucesso académico são o reforço dos meios de apoio social (bolsas de estudo); mecanismos de integração, acompanhamento e orientação; uma boa relação entre estudantes e professores; e boas infraestruturas. A DGES aponta que os fatores de abandono escolar se devem, essencialmente, ao sistema de ingresso, dificuldades económicas, expectativas goradas e a não colocação na primeira opção (Queiroz, 2015). Por seu lado, um estudo na Universidade de Aveiro (UA) refere que os estudantes com menor aproveitamento escolar no 1.º ano, ou seja, os que têm mais de 50% de ECTS em atraso, são tendencialmente estudantes inscritos em mestrado integrado ou em cursos TESP, estudantes que entram por outros regimes e estudantes de áreas das Engenharias, Ciências, Matemática e

Informática. Este estudo confirma como indicadores de insucesso as faltas às aulas, o não pagamento das propinas, a entrada com classificação menor que 120 pontos, a entrada em 5^a. e 6^a. opção e a entrada na 2^a e na 3^a fase, e identifica o PT-UA como ferramenta relevante na prevenção do insucesso (Moreira, n.d.).

Na UA existem várias iniciativas realizadas no contexto do Programa de Tutoria³, iniciado no ano letivo 2010/2011. Uma delas é o Avila Crew, iniciativa iniciada em janeiro de 2017, que visa combater o insucesso escolar na Unidade Curricular (UC) de Laboratório Multimédia 4 da licenciatura em Novas Tecnologias da Comunicação (NTC), através da realização de várias sessões ao longo do semestre, onde os estudantes participam em desafios e trocam experiências e dúvidas com ex-estudantes (mentores). Utilizam o *Slack*⁴ para permitir o diálogo entre os estudantes, mentores e tutores nos canais Slack da UC e do Avila Crew. A divulgação dos eventos é realizada através do *Facebook*. Os eventos têm como tema principal assuntos muito específicos relacionados com os conteúdos lecionados na UC, estimulando a aprendizagem num contexto de descontração, onde os estudantes realizam vários desafios/jogos, e convivem fora do contexto de uma sala de aula. Os inquéritos de satisfação realizados na primeira edição do Avila Crew revelaram que a grande maioria dos estudantes participantes acharam a iniciativa agradável e útil para o sucesso na UC, além do divertimento proporcionado e partilha de conhecimento. Os mentores consideraram um desafio organizar todos os eventos e toda a camada de gamificação, e uma oportunidade de puderem mostrar os seus pontos de vista sobre os conteúdos. O Avila Crew trouxe melhorias nos resultados da UC (Santos, 2017).

Dondero (1997), como citado em Haber-Curran et. Al (2017), refere que o mentor é uma pessoa de confiança e que toma conta de alguém em necessidade. Crisp e Cruz (2009) afirmam que os programas de mentoria nas universidades auxiliam os estudantes nas suas transições, sucesso e evitam o abandono escolar. De acordo com Beltman e Schaeben (2012) e Harmon (2006), estes programas ajudam os estudantes a desenvolver competências necessárias para o sucesso e a obtenção do sentido de pertença. Alguns programas de mentoria focam-se em estudantes considerados de risco de abandono escolar, como estudantes com deficiências, com baixos rendimentos e estudantes do primeiro ano (Haber-Curran et al., 2017).

Beltman e Schaeben (2012) constataam que uma relação entre o mentor e o estudante deve ser recíproca, na qual ambos beneficiem da relação. Os mentores ganham autoconfiança, sentem-se úteis ao servirem uma causa maior, adquirem novas perspetivas, desenvolvem competências organizacionais e competências comunicacionais (Gilles & Wilson, 2004; Beltman & Schaeben,

³ <http://www.ua.pt/tutoria/page/18959>.

⁴ O Slack (<https://slack.com/>) é uma plataforma utilizada por equipas, permitindo que estas comuniquem entre si, partilhando vários serviços e recursos, facilitando os processos.

2012; Harmon, 2006; Hughes et al., 2010; Haber-Curran et al., 2017). No entanto, estes autores salientam que, para potenciar o sucesso dos programas de tutoria, é necessário comunicar de forma adequada nos meios de comunicação mais utilizados pelos estudantes. Um programa só tem realmente utilidade quando os seus destinatários têm conhecimento, atempadamente, do mesmo para que possam usufruir.

As redes sociais são um ótimo meio para manter contactos duradouros entre mentores e estudantes, e uma forma de complementar o contacto presencial. Quando se escolhe integrar uma rede social num programa de tutoria já existente, é preciso ter o cuidado de escolher uma rede social em que tanto os estudantes, como tutores e mentores, se sintam confortáveis e confiantes. É indispensável também escolher uma rede social com base no perfil, tipo de conteúdos e objetivos pretendidos (Briones & Janoske, 2013). Para melhor se compreender como evoluíram estes e outros serviços na *web*, e em que medida dão resposta às necessidades dos utilizadores, aborda-se a seguir a evolução da Internet e como esta possibilitou o desenvolvimento de *softwares* de apoio aos utilizadores em geral, e aos programas de tutoria, em particular.

2.1.2 Evolução dos serviços e plataformas de utilização

A *web* emergiu no início dos anos 90, permitindo publicar e aceder a documentos *online*, sendo estes ficheiros transferidos por *Hypertext Transfer Protocol* (HTTP) e lidos pelo *Hypertext Markup Language* (HTML) (Hall & Tiropanis, 2012). O HTTP é um protocolo que é generalista, sem estado e utilizado para várias tarefas, permitindo que novos sistemas sejam construídos independentemente dos dados que estão a ser transferidos (Connolly, 2004). O HTML é uma linguagem base, da qual outras linguagens *web* podem a utilizar para gerar conteúdo e disponibilizá-lo aos visitantes das páginas (W3C, n.d.).

O aumento do número de linhas de telefone ajudou no crescimento da *web*, tendo em conta que a Internet nos seus primórdios era disponibilizada através das linhas telefónicas. A internet é uma forma de as pessoas publicarem informações: pessoais ou de empresas (Hall & Tiropanis, 2012).

No início do século XXI, as pessoas começaram a ter um papel mais importante na evolução da *web*, através do sucesso do *e-commerce*, que possibilitou o crescimento de novos modelos de negócio. Os utilizadores começaram a ter acesso a informações, como saberem quais os produtos mais populares, e *reviews* dos produtos feitos por outros utilizadores (Hall & Tiropanis, 2012). Surgiu a *Wikipédia*, permitindo às pessoas contribuir para o conhecimento e acesso *online* de forma gratuita. O aumento da largura de banda e das tecnologias dos servidores possibilitou o surgimento de empresas, como o Youtube (Hall & Tiropanis, 2012).

Surgiram as primeiras redes sociais *online*, permitindo às pessoas partilhar os seus interesses, ter o seu perfil *online* e estabelecer novas amizades, tendo-se atualmente como exemplo de redes sociais

online o Google Plus, LinkedIn, Facebook, etc. (Hall & Tiropanis, 2012). A Tabela 1 sintetiza as funcionalidades de algumas redes sociais utilizadas em Portugal, atualmente. Para a elaboração da Tabela 1 foram criadas contas de utilizador, de forma a possibilitar uma análise das suas funcionalidades e características.

Características \ Redes Sociais	Facebook	Twitter	LinkedIn	Instagram	Google Plus
Mensagens limitadas a 140 caracteres		x			
Focada no mundo profissional			x		
Criar páginas pessoais e de empresa	x				
Utilização de Circles para partilhar conteúdos específicos com pessoas específicas					x
Utilizadores premium conseguem saber + estatísticas sobre os visitantes e alcance das suas publicações			x		
Chat privado	x	x	x	x	x
Chamadas de vídeo	x				x
Editar dados da página de perfil	x	x	x	x	x
Recomendações de novos utilizadores	x	x	x	x	x
Partilhar ideias sobre a forma de texto	x	x	x		x
Partilhar ideias sobre a forma de imagens e vídeo	x	x	x	x	x
Aderir a comunidades	x				x
Criar comunidades	x				x

Tabela 1 – Funcionalidades das principais redes sociais utilizadas em Portugal.

Tanto o Facebook, o Google Plus, o LinkedIn, como o Twitter permitem que os utilizadores partilhem as suas ideias sob a forma de texto, imagens e vídeo (Weerasinghe, Marasinghe, Ranaweera, Amarakeerthi, & Cohen, 2013). O Instagram apenas permite partilhar ideias através de imagens e vídeo. Todas são plataformas de comunicação que permitem aderir ou criar redes de utilizadores (Rivera, 2010). Possibilitam o *chat* privado (conversação entre duas ou mais pessoas, realizada com recurso a uma ferramenta que permite comunicar em tempo real através da Internet) com outros utilizadores. Recomendam novos utilizadores, tendo como base sistemas de recomendação (Symeonidis & Tiakas, 2014). Os utilizadores podem ter a sua página de perfil e editar as suas informações pessoais e profissionais. O Facebook e o Google Plus possibilitam a chamada de vídeo. O Facebook e o Instagram permitem criar anúncios segmentados por público-alvo e *budget* pré-definido.

2.1.3 Linguagens e software de desenvolvimento

Inicialmente, as páginas *web* eram muito simples, sem qualquer estilo, sendo construídas exclusivamente em HTML (Figura 1). O HTML continua a ser utilizado, atualmente, como base

estruturante de qualquer página *web*. Com a evolução dos serviços e plataformas, aumentou o grau de exigência dos utilizadores, pelo que foi criado o *Cascade Style Sheet* (CSS) (Figura 2) para tornar os conteúdos mais legíveis, apelativos e personalizáveis. Quando o CSS foi desenvolvido, havia diferenças no aspeto das páginas dependente do tipo de *browser* utilizado, não havendo ainda uma normalização (Bos, 2016). Atualmente, o CSS criado é compatível com um grande conjunto de dispositivos e *browsers*, facilitando o processo de desenvolvimento. O CSS permite adicionar e modificar vários parâmetros, como fontes de letra, cores, espaçamentos, sendo também utilizado em *ebooks*.



Figura 1 – Primeira página *Web*, apenas com HTML (Arah, 2005)⁵.

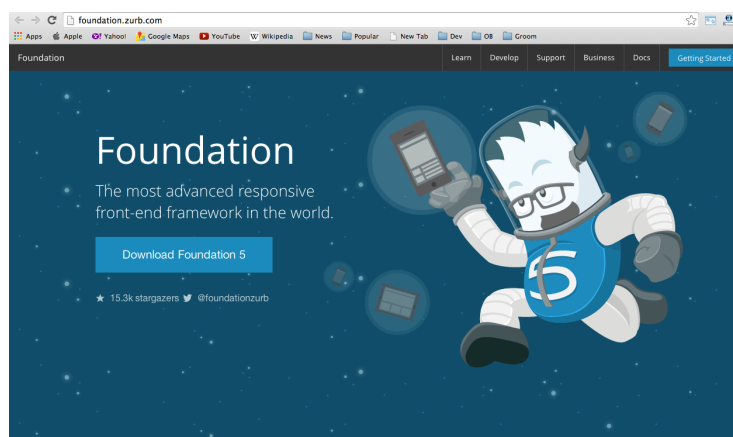


Figura 2 – Exemplo de uma página *Web* com CSS (ieatcss.com, n.d.)⁶.

⁵ Imagem retirada do website http://designer-info.com/Web/cms_introduction.htm, em 19/01/2018.

O *JavaScript* (Figura 3) surgiu em maio de 1995 por um colaborador da Netscape (atual Mozilla Firefox), com o objetivo de trazer maior animação e interação dos utilizadores através do *Document Object Model* (DOM) (w3.org, 2017). A Microsoft tinha uma versão do *JavaScript* para o *browser* Internet Explorer, o que obrigava os programadores a desenvolver versões diferentes consoante o *browser*. Assim, foi criada uma normalização: o *ECMAScript*. Inicialmente, o *Javascript* apenas funcionava no interior do *browser*, não existindo qualquer forma de haver *server side rendering*, nem a possibilidade de correr programas *Javascript* pelo terminal da linha de comandos. Quando a *Google* criou o *NodeJS*, o *Javascript* passou a ser utilizado no lado *server side* e na linha de comandos, trazendo um conjunto de novas funcionalidades (w3.org, 2017).

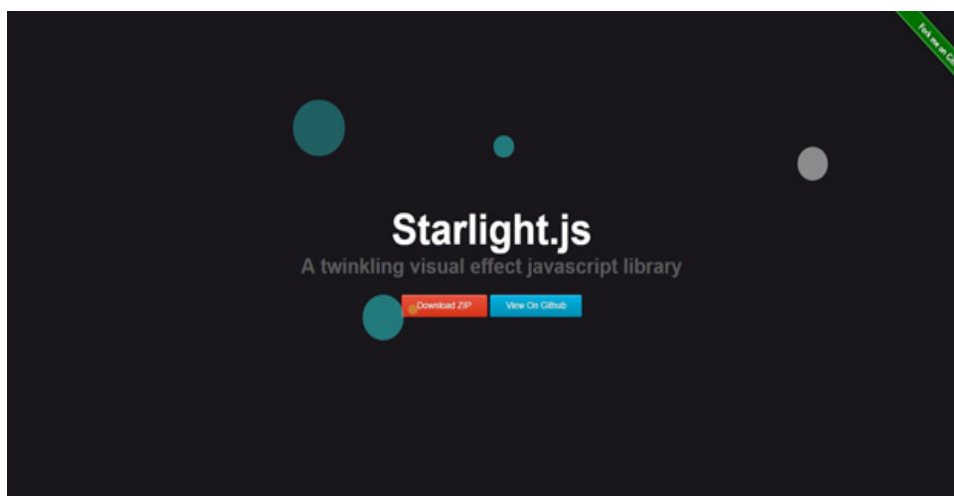


Figura 3 – Biblioteca *Javascript* que adiciona efeitos visuais (Jqueryhouse.com, n.d.)⁷.

À medida que as necessidades de desenvolvimento se tornam mais exigentes, surge a vontade de acrescentar novas funcionalidades nas linguagens, para que seja possível construir páginas *web* de forma mais fácil e robusta. Essas novas funcionalidades são propostas à comunidade por detrás do desenvolvimento do *Javascript*, para que estas sejam testadas e implementadas.

Quando ocorre a proposta de novas alterações, esta passa por um conjunto de quatro estágios. No primeiro estágio é apresentado um conjunto de exemplos de problemas que esta proposta irá resolver e os desafios apresentados. Se a proposta for considerada importante, segue-se o segundo estágio, onde deverá apresentar a sintaxe e semântica das novas funcionalidades. No terceiro estágio, a proposta é testada pelos implementadores e utilizadores, fornecendo *feedback*. No quarto estágio, a proposta está pronta a ser implementada nas especificações oficiais (McGinnis, n.d.).

⁶ Imagem retirada do website <http://ieatcss.com/responsive-web-design/css-frameworks/zurb-foundation-css-framework.html>, em 19/01/2018.

⁷ Imagem retirada do website <https://jqueryhouse.com/javascript-animation-libraries/>, em 19/01/2018.

Têm surgido várias versões do *ECMAScript*, sendo que as mais recentes necessitam de ser compiladas para o *ECMAScript5*, que é o único que o *browser* consegue interpretar. No *ECMAScript5* (ES5) existia apenas um único tipo de variável (*var*), que provocava alguns problemas devido ao facto da *scope* do objeto estar acessível fora do local onde era invocado, passando no *ECMAScript6* (ES6) a ter três tipos: *var*, *let* e *const* (Johansson, 2016). O *const* apenas é usado para variáveis cujo valor não irá sofrer alterações, o *let* é usado em situações em que o *scope* apenas é pretendido dentro da função onde é definida e o *var* quando se pretende que a variável tenha *scope* global (Figura 4). No ES6 é possível importar módulos com o *import*, classes e heranças muito úteis para a programação orientada a objetos, e *string templates* – que permitem interpolar pelo programador (Quora, 2016).

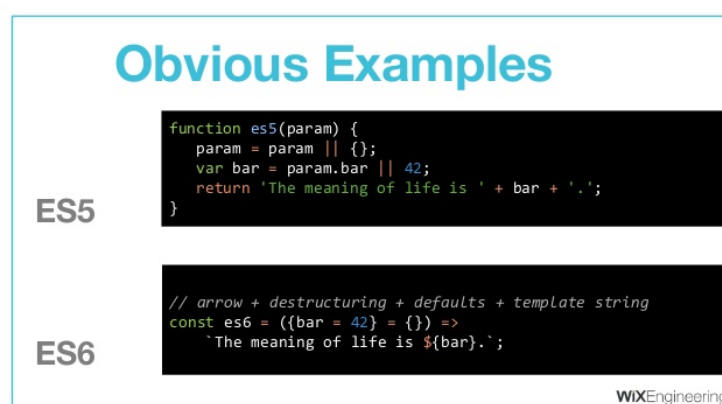
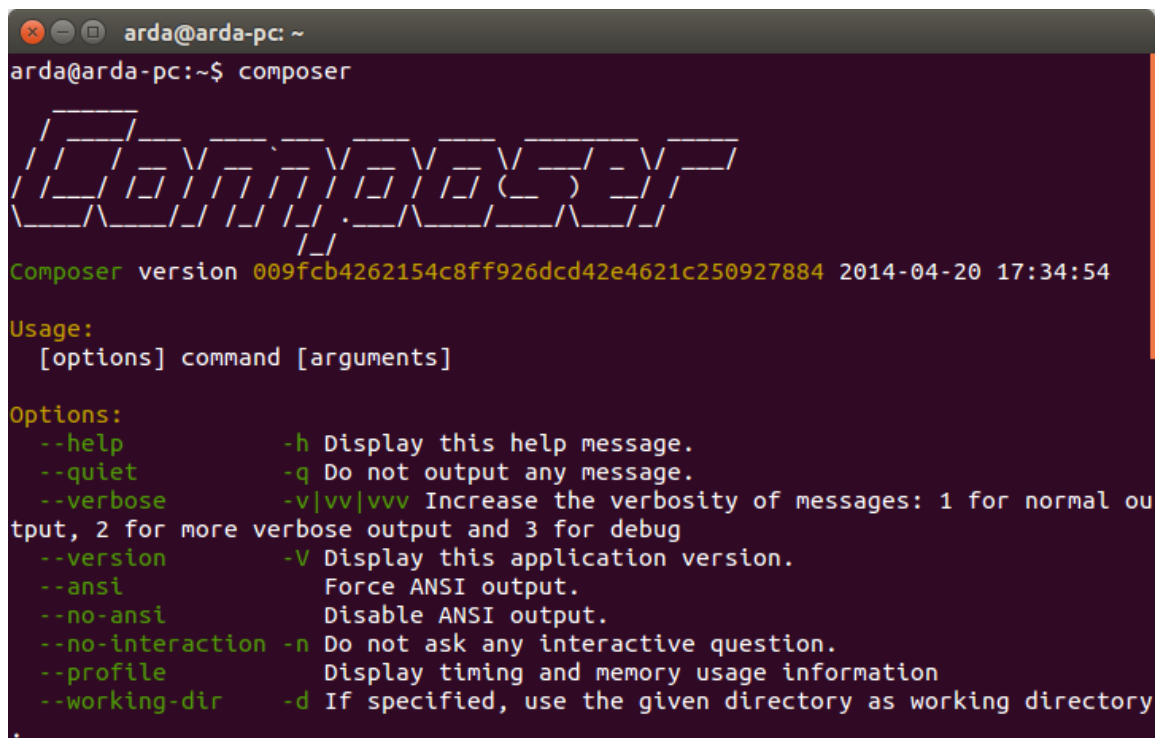


Figura 4 – Diferenças de sintaxe do ES5 para o ES6 (Shappir, 2016)⁸.

Outra linguagem amplamente utilizada no desenvolvimento *web* é o PHP (*Hypertext Preprocessor*), sendo esta uma linguagem de *script open source* embutida dentro do HTML (The PHP Group, 2018). O PHP é utilizado para gerar páginas *web* com conteúdo dinâmico, obter dados dos formulários e inserir os dados nas bases de dados, permite enviar e receber *cookies*. Pode ser utilizado para gerar *scripts* nas linhas de comando, fazendo com que este possa ser executado fora do contexto *web*, dando a possibilidade de desenvolver aplicações *desktop* ou até mesmo rotinas de processamento de texto simples. Suporta uma ampla variedade de bases de dados, facilitando todo o processo de comunicação (Figura 5).

⁸ Imagem retirada do website <https://www.slideshare.net/DanielleAVincent/dan-shappir-things-you-can-do-in-es6-that-cant-be-done-in-es5>, em 19/01/2018.



```
arda@arda-pc: ~
arda@arda-pc:~$ composer

Composer version 009fcb4262154c8ff926dcd42e4621c250927884 2014-04-20 17:34:54

Usage:
[options] command [arguments]

Options:
--help           -h Display this help message.
--quiet          -q Do not output any message.
--verbose        -v|vv|vvv Increase the verbosity of messages: 1 for normal ou
tput, 2 for more verbose output and 3 for debug
--version        -V Display this application version.
--ansi           Force ANSI output.
--no-ansi        Disable ANSI output.
--no-interaction -n Do not ask any interactive question.
--profile        Display timing and memory usage information
--working-dir    -d If specified, use the given directory as working directory
```

Figura 5 – Exemplo da utilização do PHP pela linha de comandos (Arda, 2014)⁹.

O desenvolvimento de *software* tem também sofrido alterações e novos sistemas/paradigmas cada vez mais complexos. Tornou-se necessário dividir o problema em várias partes, separando as funcionalidades dos acessos aos bancos de dados e da apresentação (número de visualizações). Este princípio é conhecido por “separação de preocupações” (*separation of concerns*), contribuindo para que o *software* seja mais fácil de gerir e de atribuir responsabilidades (Kumar, Kumar, & Iyyappan, 2016). Inicialmente, o paradigma dominante era a Programação Estruturada (PE), que foi substituída pela Programação Orientada e Objetos (POO). Na PE havia muita repetição de código, violando o princípio DRY (*Don't Repeat Yourself*) e não existiam classes nem a possibilidade de herdar atributos e comportamentos. A PE, por sua vez, permite um melhor controlo do fluxo de execução do código, e uma compreensão mais fácil, sendo amplamente utilizada em cursos de introdução à programação. A PE tem tendência a gerar códigos confusos e a misturar no mesmo ficheiro componentes e lógica com bases de dados e apresentação (Kumar et al., 2016).

Atualmente, o paradigma dominante é a POO, onde os programas são compostos por objetos divididos por classes e que possuem um conjunto de atributos e comportamentos que podem ser partilhados/herdados por outros objetos. O POO aplica o princípio DRY, permitindo uma melhor reutilização, extensibilidade e programação mais eficiente (Dong, 2011). A Programação Orientada

⁹ Imagem retirada do website <https://serverfault.com/questions/591105/centos-colors-in-php-cli-applications>, em 19/01/2018.

a Aspetos (POA) surgiu recentemente e permite desenvolver um conjunto de módulos funcionais que podem ser incorporados na componente lógica quando necessário. Permite uma modularização e reutilização do código (Dong, 2011).

De forma a conseguir responder rapidamente às necessidades do mercado, as *startups*, numa fase embrionária, desenvolvem o seu código de forma rápida e ineficiente. À medida que vão crescendo, sentem a necessidade de refazer o código devido à sua falta de escalabilidade, a qual diz respeito à capacidade de um código ser ajustável ao tamanho do sistema. A escalabilidade é medida através de diferentes dimensões, como a capacidade de aguentar mais dados, maiores níveis de concorrência e maiores interações (Ejsmont, 2015a). A escalabilidade está relacionada com a *performance*. A *performance* de um sistema mede quanto tempo demora a desempenhar uma certa tarefa, enquanto a escalabilidade mede quanto um sistema pode ser aumentado sem ser necessário mudar o código. Existem várias etapas na escalabilidade de uma *startup*, desde os mais simples como um único servidor a *data centers* e serviços como o *Amazon EC2*. À medida que os negócios aumentam é preciso gerir mais dados (mais contas de utilizadores, mais produtos), o que provoca muita tensão no sistema. Os maiores níveis de concorrência medem a capacidade de um sistema servir um maior número de clientes ao mesmo tempo. As maiores interações medem a capacidade de o sistema aguentar o aumento de trocas de informação entre os servidores e os utilizadores (Ejsmont, 2015a).

Num sistema com um único servidor, os utilizadores fazem uma conexão com o DNS (*Domain Name System*) para obterem o IP (*Internet Protocol*) do endereço onde está alojado o *website*, e após a obtenção do IP são feitos os pedidos HTTP ao servidor. A desvantagem deste sistema é que apenas uma máquina é responsável por toda a carga necessária para o *website* funcionar, tornando-se ineficiente quando o volume de tráfego e dados aumenta. Quando esse sistema chega à sua rutura, é necessário tomar novas decisões, havendo soluções de escalabilidade vertical e horizontal (Ejsmont, 2015a). Existem várias formas de escalar verticalmente, como adicionar mais discos em RAID (*Redundant Array of Independent Disks*); trocar os discos HDD por discos SSD (*Solid-State Drive*), permitindo leituras e escritas de dados mais rápidas entre 10 a 100 vezes; aumentar a RAM, possibilitando mais memória para o funcionamento da aplicação; fazer a troca para servidores que possuam mais processadores. A escalabilidade vertical é uma ótima opção para pequenas aplicações ou para empresas que consigam suportar *upgrades* de *hardware*. Uma das desvantagens é o facto de terem limites, não sendo possível aumentar memória infinitamente. O *design* do sistema operativo e da aplicação podem impedir a escalabilidade vertical a partir de um certo ponto de crescimento (Ejsmont, 2015a). Outras soluções passam por isolar os vários componentes/serviços da aplicação em vários servidores, por exemplo, um servidor para a base de dados e outro para o Apache/NGINX. A utilização de CDN (*Content Delivery Network*) permite que os dados fiquem em cache nos servidores de CDN, fazendo com que os pedidos dos clientes

deixem de ser feitos ao nosso servidor, reduzindo a carga no sistema (Ejsmont, 2015a). A escalabilidade horizontal permite aumentar a capacidade do sistema através de um conjunto de métodos que adicionam mais servidores à medida que são necessários. A partir de um certo ponto, a escalabilidade horizontal é mais vantajosa em termos de benefício/custo, devido em certos serviços, como o da Amazon CloudFont em que o preço reduz com o aumento da escalabilidade. Quando uma *startup* chega a ter milhões de utilizadores espalhados pelo mundo, passa a ter uma necessidade de ter *data centers* espalhados por várias regiões do mundo para reduzir a latência. A utilização de *load balancers* permite direcionar o tráfego de um *website* pelos vários servidores, escolhendo o que se encontra mais próximo fisicamente do cliente (Ejsmont, 2015a). A camada de aplicação *web* não deve conter a lógica de funcionamento e limitar-se a renderizar as interfaces dos utilizadores e delegar a lógica de funcionamento para a camada dos serviços *web*. A camada de serviços *web* é constituída por um conjunto de serviços autónomos que realizam e contém uma lógica muito específica.

De acordo com Ejsmont (2015b), alguns dos princípios de bom *design* de *software* são: simplicidade; esconder a complexidade e construir abstrações; evitar *Overengineering*; *Test-Driven Development* (TDD); aprender de outros modelos de *design* de *software* simples; reduzir o acoplamento; *Don't Repeat Yourself* (DRY); desenhar diagramas; atribuir responsabilidades e *Dependency Injection*. Seguidamente serão explicados cada um destes princípios:

- Simplicidade: desenvolver *software* simples é difícil de mensurar, uma vez que é um conceito subjetivo. Um *software* simples é fácil de compreender e utilizar, mesmo que seja desenvolvido por um engenheiro externo à empresa. Este conceito é muito importante a médio e longo prazo, quando são contratados novos elementos para uma equipa e estes necessitam de utilizar e compreender.
- Esconder a complexidade e construir abstrações: à medida que um sistema cresce, é cada vez mais difícil manter um mapa mental do mesmo, sendo necessário construir módulos que abstraíam. O engenheiro de *software* não necessita de ver a implementação de todos os módulos, mas apenas compreender quais as tarefas que esses executam para conseguir ter um mapa mental da aplicação.
- Evitar *Overengineering*: os engenheiros têm tendência a desenvolver *software* complexo. Deve ser desenvolvido software o menos complexo possível, garantindo as funcionalidades importantes e de forma a que seja relativamente fácil acrescentar mais funcionalidades posteriormente.
- *Test-Driven Development* (TDD): a metodologia TDD promove a simplicidade, porque obriga a pensar no ponto de vista de utilização do utilizador enquanto se constrói os testes.

- Aprender de outros modelos de *design* de *software* simples: a *Framework Grails* e a API do Google Maps são exemplos de modelos simples de *design* que permitem a sua adoção e compreensão de forma rápida e sem grandes dificuldades.
- Reduzir o acoplamento: reduzir o acoplamento entre os vários componentes do sistema facilita a escalabilidade, uma vez que estes não têm conhecimento dos restantes elementos nem dependem destes para o seu funcionamento.
- *Don't Repeat Yourself* (DRY): repetir o mesmo código vezes sem conta é desperdiçar tempo e recursos que poderiam estar a ser utilizados noutros locais importantes. É necessário automatizar os processos e reduzir o número de processos ineficientes.
- Desenhar diagramas: quando se desenham diagramas está a sintetizar-se conhecimento e facilita a compreensão e partilha com outros engenheiros.
- Atribuir responsabilidades: este princípio defende que uma classe deve ter apenas uma responsabilidade, aumentando a simplicidade, reutilização e teste de código desenvolvido.
- *Dependency Injection*: a *dependency injection* reduz o acoplamento e torna o código mais simples, uma vez que a classe não necessita de saber quem é ou quem implementa determinadas funcionalidades, necessitando apenas de se focar na sua única responsabilidade.

Os softwares de apoio a programas de tutoria, conhecidos atualmente, utilizam tecnologias baseadas na web, como o Javascript, PHP, HTML e CSS para a utilização em computador pessoal, e linguagens nativas para as versões *App*, como é o caso do software Achiiva (Achiiva, 2016).

No próximo capítulo apresentam-se o conceito de aplicação móvel (vulgarmente abreviada por *App*) e as principais recomendações e *standards* do mercado. O objetivo dessa análise é perceber quais os principais critérios a ter em conta para conceber os *layouts* de uma *App* que vá ao encontro do que os utilizadores estão acostumados, assim como quais os softwares mais utilizados para esse fim. São também apresentadas as principais linguagens e *frameworks* utilizadas para a implementação de *Apps*.

2.2 Aplicação móvel

Apresenta-se a definição de aplicação móvel, assim como os princípios de desenvolvimento de *layouts*, tanto em *design*, como em código, assim como as principais ferramentas para o *design* dos mesmos. A relevância advém da necessidade de conhecer quais os *standards* do mercado, para que a aplicação do PT-UA vá ao encontro das necessidades dos seus utilizadores.

2.2.1 Definição e princípios de desenvolvimento

Uma aplicação móvel é um programa, *software* criado para ser executado em dispositivos móveis como *smartphones*, *tablets* e *smartwatches*. Uma aplicação móvel é vulgarmente conhecida pelo termo *App* e tornou-se popular quando a *Apple* criou a *App Store*. As aplicações só podem ser obtidas e instaladas através das lojas oficiais, *App Store* no caso de aparelhos iOS e *Google Play* no caso de dispositivos *Android* (Tech Terms, 2012).

Com o surgimento e desenvolvimento dos dispositivos móveis e a atual proliferação de aplicações, há uma constante luta pela atenção e pelo interesse do utilizador. Este novo perfil de utilizador vive rodeado de informação e acede a várias aplicações durante o dia, mas por curtos períodos de tempo. Assim, torna-se necessário disponibilizar conteúdos relevantes, de consulta rápida. É indispensável conhecer muito bem o público-alvo para poder acompanhar a forma como a aplicação é utilizada (Smilansky, 2016).

Antes de se começar a desenvolver o *design* de uma aplicação é importante seguir algumas recomendações, nomeadamente, de dois grandes nomes no que diz respeito ao desenvolvimento móvel: a *Apple* e a *Google*. As recomendações têm como base a sua experiência no lançamento das suas próprias *Apps* e de estatísticas referentes a outras *Apps* desenvolvidas por terceiros e disponibilizadas nas suas lojas. Ao seguir essas recomendações, há uma grande probabilidade de a aplicação ser agradável de utilizar e ter sucesso. A *Apple* apresenta um conjunto de recomendações sobre vários aspetos essenciais, desde os cuidados a ter com a escolha das cores, da tipografia a utilizar, dos elementos da interface, das animações mais apropriadas, e o tipo de conteúdo adequado, etc. Estas facilitam todo o trabalho de *design* de uma aplicação, e aumentam a probabilidade de esta ser adequada ao seu público e atingir o sucesso. A *Apple* tem interesse em auxiliar os *designers* e programadores, uma vez que além de receberem dinheiro das licenças de desenvolvedor, a existência de uma quantidade considerável de aplicações de qualidade potencia as vendas dos seus produtos. As principais recomendações são apresentadas a seguir, sob a forma de tópicos, para facilitar a leitura e compreensão.

Algumas das recomendações gerais da Apple Inc. (2017w) são:

- Criar *layouts* adaptáveis aos ecrãs dos dispositivos, para que o utilizador não tenha de fazer *zoom* ou *scroll* horizontal.
- Utilizar elementos User Interface (UI) que permitam os movimentos de toque para facilitar a interação e esta ser mais fluída e natural.
- Os botões de navegação deverão ter dimensões que favoreçam o *touchscreen*.

- O tamanho do texto deverá ser legível a uma distância visual razoável, sem que seja necessário fazer *zoom*.
- Deve haver contraste entre a cor do texto e o fundo e um espaçamento entre linhas razoável, para que o texto seja legível.
- Providenciar imagens de várias resoluções, para que estas não apareçam desfocadas em ecrãs retina e manter o rácio nas proporções das imagens, para que estas não fiquem esticadas.
- Criar *layouts* fáceis de interagir, com botões de navegação funcionais

Em relação aos *layouts*, é importante seguir as seguintes recomendações (Apple Inc., 2017c):

- Numa aplicação é necessário dar prioridade ao conteúdo.
- As pessoas não devem ter de fazer *scroll* horizontal ou zoom para conseguirem ler os conteúdos.
- Manter uma consistência visual ao longo de toda a aplicação.
- Os itens mais importantes devem ser maiores que os restantes, de forma a se destacarem.
- Os conteúdos devem ser alinhados de forma organizada e hierarquicamente.
- A aplicação deverá ocupar grande parte do ecrã.
- Deve garantir-se espaçamento e margens adequadas ao longo do conteúdo. Aplicar margens que facilitem a leitura do texto em ecrãs maiores.
- Os botões de navegação deverão ter dimensões adequadas para facilitar o toque.
- Quando uma pessoa roda um dispositivo, o *layout* não tem de mudar. Se possível, ter um *layout* para *portrait* e para *landscape*.
- Animações (Apple Inc., 2017e):
 - Utilizar animações de forma ponderada e com contexto.
 - Procurar o realismo e credibilidade.
 - Usar animações consistentes.
- *Branding* (Apple Inc., 2017g):
 - O *branding* não deve ser intrusivo.
 - Não deve estar à frente de um grande design.

- Dar preferência ao conteúdo em detrimento do *branding*.
- Resistir à tentação de apresentar o *logo* ao longo de toda a aplicação.
- Cor (Apple Inc., 2017i):
 - Escolher cores indicadas para os objetivos de comunicação.
 - Usar cores complementares ao longo de toda a aplicação.
 - Escolher uma cor chave para representar a interatividade ao longo de toda a aplicação.
 - Não misturar a mesma cor utilizada para representar a interatividade com elementos não interativos.
 - Testar as cores em várias condições de luz.
 - Usar o contraste de cores necessário para facilitar a leitura.
- Terminologia (Apple Inc., 2017s):
 - Utilizar palavras e frases familiares que sejam facilmente compreendidas pelo público-alvo.
 - Manter a interface limpa e concisa.
 - Facilidade em detetar quais as funcionalidades que cada elemento realiza.
 - Usar um tom de comunicação informal e amigável.
 - Ter cuidado ao utilizar humor.
- Tipografia (Apple Inc., 2017v):
 - Dar ênfase à informação mais importante.
 - Se for possível, utilizar apenas um tipo de letra.
 - Se forem adicionadas fontes de letra personalizadas, é necessário garantir que são legíveis.
- App Icon (Apple Inc., 2017f):
 - Os ícones deverão ser simples, com um único ponto focal, ser facilmente reconhecíveis e evitar usar fundos transparentes.
 - Manter os cantos dos ícones quadrados.
 - Não incluir fotos, *printscreens* ou elementos da interface.

- *Launch Screen* (Apple Inc., 2017k):
 - Criar um *launch screen* idêntico ao primeiro ecrã da aplicação.
 - Evitar usar texto.
 - Não fazer publicidade.
- Barra de Navegação (Apple Inc., 2017m):
 - Evitar colocar muitos itens na navegação.
 - Usar o botão de voltar *standard*.
- Barra de Pesquisa (Apple Inc., 2017o):
 - Utilizar uma barra de pesquisa em vez de um campo de texto.
 - Não retirar o botão de “limpar” a barra de pesquisa.
 - Permitir o botão de “cancelar”.
 - Colocar *placeholders* para auxiliar os utilizadores.
- Separadores / *tab bars* (Apple Inc., 2017q):
 - Organizar a informação através de separadores.
 - Utilizar os separadores apenas para auxiliar a navegação.
 - Garantir a consistência entre os ícones utilizados nos separadores.
- Barras de Ferramentas / *toolbars* (Apple Inc., 2017u):
 - Os botões do *toolbar* devem ser relevantes.
 - Se forem de texto deve ser dado o espaçamento adequado.
- Alertas de ação / *action sheets* (Apple Inc., 2017b):
 - Apresentar a opção de cancelar.
 - Os elementos de destruição como o apagar devem ser diferentes dos restantes.
 - Evitar o *scroll* enquanto estiverem disponibilizados os alertas de ação.
- Alertas (Apple Inc., 2017d):
 - Minimizar os alertas e testar o seu aspeto nas duas orientações.
 - Escrever mensagens curtas, evitando o tom acusatório, julgamentos ou insultos.

- Deverá ter dois botões, sendo que os de destruição deverão diferenciar-se dos restantes e devem estar localizados em sítios esperados pelos utilizadores.
- Mapas (Apple Inc., 2017l):
 - Manter os mapas interativos e com pins coloridos a identificar pontos de interesse.
- *Scroll views* (Apple Inc., 2017n):
 - Não utilizar *scroll views* dentro de outras *scroll views*.
 - Apresentar uma *scroll view* de cada vez.
- Tabelas (Apple Inc., 2017r):
 - Pensar na largura das tabelas.
 - Mostrar os conteúdos de forma imediata e comunicar o processo à medida que esta carrega.
 - O texto deverá ser sucinto.
- *Text views* (Apple Inc., 2017t):
 - O texto deverá ser bem legível e apresentar o tipo de teclado adequado.
- *Web views* (Apple Inc., 2017x):
 - Disponibilizar botões de “avançar” e “voltar” na navegação quando apropriado.
 - Evitar utilizar uma *web view* para construir um *web browser*.
- Botões (Apple Inc., 2017h):
 - Os títulos dos botões deverão ser verbos com a primeira letra maiúscula e curta.
 - Adicionar um contorno ou fundo apenas se necessário.
- *Labels* (Apple Inc., 2017j):
 - Os *labels* deverão ser legíveis e geralmente as pessoas copiam os seus conteúdos, embora não os possam editar.
- *Sliders* (Apple Inc., 2017p):
 - Não utilizar *sliders* para ajustar o volume de áudio e apenas modificar o seu aspeto se isso acrescentar valor para os utilizadores.

A Google desenvolveu o *material design* tendo em conta bons princípios de *design*, que permite uma experiência semelhante ao longo das plataformas e aparelhos. Adota como princípios a

metáfora do material, a hierarquia e a ênfase no mais importante, e defende que só se deve usar o movimento se este tiver significado (Apache 2, n.d.-k). O *material design* diz respeito a um ambiente tridimensional que contém luz, materiais e sombras. No mundo real, os objetos 3D têm os eixos X, Y e Z, enquanto na Web o material tem apenas o X e Y, sendo necessário emular o 3D, manipulando o valor de Y. A fonte e direção de luz traz influência no tipo de sombra (Apache 2, n.d.-c). O *material design* foi inspirado nas características dos materiais reais para criar *layouts* consistentes, de acordo com a percepção do ser humano do mundo real (Apache 2, n.d.-i). No mundo real, os objetos não podem passar uns pelos outros e o mesmo se aplica ao *material design*. Numa aplicação, a elevação é a distância entre duas superfícies. Quando clicamos no menu de uma aplicação, este desliza, ficando na superfície de outras camadas (Apache 2, n.d.-j). A hierarquia dos objetos é feita com base na relação que se estabelece entre eles, geralmente de componente “pai” para componentes “filho”. Cada objeto tem apenas um pai, podendo ter vários filhos. Os filhos herdam propriedades dos pais. Segundo a *Google*, o movimento permite guiar o foco dos utilizadores ao longo dos ecrãs, relações hierárquicas e espaciais entre elementos (Apache 2, n.d.-h). Os movimentos são usados para criar transições entre ecrãs e para dar *feedback* da utilização aos utilizadores (Apache 2, n.d.-f). Ao criar animações é necessário definir a sua duração e se esta irá ser mais suave ou abrupta nas transições (Apache 2, n.d.-b). As transformações do material permitem dar vida aos ecrãs através da multiplicação, divisão, tamanho e forma dos objetos (Apache 2, n.d.-g).

No *material design*, a cor primária é a que aparece mais frequentemente na aplicação e a cor secundária é utilizada para dar importância a certos elementos da *User Interface* (UI). Só se deve utilizar textos coloridos e fundos coloridos para elementos textuais realmente importantes (Apache 2, n.d.-a). Utilizam-se figuras geométricas para representar ideias-chave, capacidades ou serviços. Os ícones de produtos deverão ter 48 *Density Pixels*(dp) e ícones de sistema 24 dp (Apache 2, n.d.-d). As imagens são uma ótima forma de comunicar de forma diferenciada, desde que sejam agradáveis, elegantes e informativas. É preciso ter em atenção a forma como estas se integram com os restantes elementos da UI e com a sua resolução. Devem usar-se imagens específicas sem ser de bases de imagens e haver um único ponto de foco, de forma a conseguir construir uma narrativa (Apache 2, n.d.-e). A *Google* defende que os textos devem ser fáceis de ler e compreender por qualquer pessoa (Apache 2, n.d.-l).

A escolha da cor a utilizar é muito importante, uma vez que a mesma tem o poder de expressar emoções e simbolizar ideias, e o seu significado pode variar dentro da mesma cor, ao modificar a sua tonalidade e intensidade. As cores têm significados positivos ou negativos dependendo do seu contexto de utilização. A interpretação do significado da cor pode variar consoante a idade, estado de espírito, experiência pessoal, história e tradição.

A cor vermelha é ligada a elementos como o sangue e o fogo. Pelo lado positivo é associada ao amor, à energia, ao entusiasmo e ao poder. No lado negativo é associada à revolução, à agressão, à raiva e à batalha. O amarelo é associado ao brilho do sol. Pelo lado positivo é associado ao idealismo, ao otimismo e ao intelecto. Pelo lado negativo pode ser associado à ganância, à covardia e ao engano. O azul é associado ao céu e ao oceano. Pelo lado positivo é associado ao conhecimento, à paz, à lealdade e à justiça. Pelo lado negativo é associado à depressão e à apatia. O verde é ligado ao ambiente e à natureza. Pelo lado positivo é associado ao dinheiro, à fertilidade, ao sucesso, ao crescimento, à harmonia e à juventude. Pelo lado negativo é associado à ganância, ao veneno, à corrosão e à falta de experiência. O roxo é ligado à realeza e à espiritualidade. Pelo lado positivo é associado ao conhecimento, à imaginação, à saúde, ao misticismo, à luxúria e à mobilidade. Pelo lado negativo é associado ao exagero, ao excesso, à loucura e à crueldade. A cor de laranja é associada ao outono e às laranjas. Pelo lado positivo transmite sensações de criatividade, energia, saúde e sociabilidade. Pelo lado negativo é associado ao barulho e à moda. O preto é associado à noite e à morte. Pelo lado positivo simboliza o poder, a autoridade, a elegância e a sofisticação. Pelo lado negativo simboliza o medo, o secretismo, o vazio, o negativismo e a submissão. A cor branca é associada à luz e à pureza. Pelo lado positivo simboliza a perfeição, a inocência, a pureza, a verdade e a simplicidade. Pelo lado negativo simboliza a fragilidade e o isolamento. A cor cinzenta simboliza a neutralidade. Pelo lado positivo é o equilíbrio, a segurança, a maturidade e a inteligência. Pelo lado negativo é a incerteza, a indecisão, a tristeza e o mau tempo (Stone, Adams, & Morioka, 2006).

Os tipos de letra são fundamentais na comunicação visual, existindo muitas variedades de tipos de letra, desde os mais simples e legíveis aos mais abstratos e artísticos (Cullen, 2012). É preciso ter bastante cuidado a escolher tipos de letra, tendo em conta que as tipografias são apresentadas de maneira diferente consoante as plataformas e *browsers*. Os tipos de letra criam atmosfera, pelo que é preciso escolher um tipo de letra adequado ao que se pretende comunicar (Cullen, 2012). Podem ser agrupadas em super-famílias, incluindo fontes com serifa, semi-serifa, sem serifa, *semi sans*, etc. O *designer* deve fazer testes aos tipos de letra em vários dispositivos para escolher os que se enquadram melhor (Cullen, 2012).

O processo de desenvolvimento dos *layouts* da *App* PT-UA tem em consideração as recomendações apresentadas, de forma a mitigar a probabilidade dos *layouts* não se adequarem aos seus utilizadores e a sua experiência de utilização ser medíocre. No próximo subcapítulo são apresentadas algumas ferramentas para o *mockup* de aplicações.

2.2.2 Ferramentas para *mockup*

No mercado existem muitos *softwares* para o desenvolvimento de *mockups*, os *quais* podem ser desenvolvidos manualmente ou em *software* apropriado, existindo alternativas gratuitas/*open source* e pagas. Os *mockups* podem ter o formato de *wireframing* mais abstratos ou mais realistas, com utilização de imagens e com uma maior fidelidade.

O *software* Balsamiq¹⁰ é baseado em *cloud* e é um dos mais conhecidos para desenvolvimento (fácil e rápido) de *mockups*. É um *software* pago e oferece vários preços consoante o tipo de licença e forma de pagamento (Balsamiq Studios, 2017).

O Mockingbird¹¹ é um *software* pago, de desenvolvimento de *wireframing* de aplicações *mobile* e *websites*. É composto por vários elementos de UI representados e que podem ser arrastados e reposicionados para obter o *layout* desejado (Some Character LLC, n.d.).

O InVision¹² permite transformar *wireframes* estáticos em elementos clicáveis e interativos, além de os tornar *live*, facilitando os testes de usabilidade após a fase de criação dos *mockups*. É gratuito para um único projeto, sendo os restantes pagos (InVision, 2017).

O Adobe XD¹³ é uma ferramenta gratuita, que permite a construção rápida de *wireframes* e *mockups*, podendo ser adicionada alguma interatividade, além da partilha *live* que facilita os testes (Adobe Systems Incorporated, 2017).

O Adobe Photoshop¹⁴ é uma ferramenta paga, a mais famosa de todos os tempos para a manipulação e edição de imagem, sendo ainda amplamente utilizada para o design de *mockups* de páginas *web* e aplicações móveis.

O Axure¹⁵ é um *software* pago, que permite uma construção simples e rápida de *mockups*, sendo possível adicionar alguma interatividade e animações (Axure Software Solutions Inc., 2017).

O Justinmind¹⁶ permite a construção rápida de *mockups*, adicionar interatividade e animação, sendo possível publicar os protótipos e receber feedback por parte dos colegas (Justinmind, 2017).

¹⁰ Consultar <https://balsamiq.com/> para encontrar mais informações do Balsamiq.

¹¹ Consultar <https://gomockingbird.com/> para mais informações do Mockingbird.

¹² Consultar www.invisionapp.com para mais informações do InVision.

¹³ Consultar www.adobe.com/pt/products/xd.html para mais informações sobre o Adobe XD.

¹⁴ Consultar <http://www.adobe.com/pt/products/photoshop.html> para mais informações sobre o Adobe Photoshop.

¹⁵ Consultar www.axure.com para mais informações sobre o Axure.

¹⁶ Consultar www.justinmind.com para mais informações sobre o JustinInMind.

A Tabela 2 sintetiza as características dos *software* para desenvolvimento de *mockups* abordados, e as informações foram retiradas das suas páginas *web*.

Software	Balsamiq	Mockingbird	Invision	Adobe XD	Photoshop	Axure	Just In Mind
Características							
Software pago	x	x	x		x	x	x
Baseado em cloud	x						
Desenvolvimento fácil e rápido	x	x	x	x	x	x	x
Wireframing estático	x	x	x	x	x	x	x
wireframing dinâmico			x	x		x	x
Acesso live (p.e: realizar testes de usabilidade)			x	x			x
Mockups de maior fidelidade				x	x		
Opção gratuita			x	x			
Manipulação de imagem					x		

Tabela 2 - Características de *software* para *mockup*.

Analisando a Tabela 2 podemos verificar que não existem grandes diferenças entre os vários *software*, tornando a tomada de decisão um assunto de escolha por preferência ou familiaridade com a utilização de alguns destes *software*. Das alternativas apresentadas apenas uma é gratuita, o Adobe XD.

Para a realização dos *mockups* é aconselhável desenhar os vários ecrãs de forma estática e posteriormente adicionar animações para os tornar dinâmicos. Tornar dinâmico é, por exemplo, adicionar transições entre ecrãs, ou o utilizador ao clicar num botão de ver comentários, apresentar os comentários, tornando a experiência o mais real possível da utilização final.

2.2.3 Formas de desenvolvimento

Apresentam-se agora as várias formas de desenvolver aplicações móveis e quais os principais tipos de bases de dados existentes. São também apresentadas algumas tecnologias que podem ser utilizadas para o desenvolvimento da versão *Web*, que serve de base para a gestão de conteúdos e utilizadores da aplicação. Para tal, distinguem-se as arquiteturas REST e SOAP. Justificam-se a opções feitas no âmbito deste trabalho.

No desenvolvimento de aplicações *móveis* existem duas abordagens principais: com recurso a linguagens nativas (*Swift/Objective-C* no caso do *iOs* e *Java* para os dispositivos *Android*) ou com recurso a tecnologias *web*, sendo posteriormente compiladas de forma a poderem ser interpretadas pelos dispositivos. Dentro do desenvolvimento com recurso a tecnologias *web* destacam-se as *frameworks* Cordova e React Native. Antes de se começar a escrever código, é necessário pensar para que sistemas operativos vamos desenvolver, uma vez que se optarmos por desenvolver em linguagens nativas, serão necessárias duas equipas e programar duas aplicações, enquanto que se

for baseado em tecnologias *web* é apenas necessária uma equipa, sendo por isso mais económico (Saleh, Holmes, Bray, & Yusuf, 2016a).

Inicialmente, o desenvolvimento para iOS baseava-se na linguagem *Objective-C*, presente há várias décadas e criada com base na linguagem C e na linguagem *Smalltalk*. O *Objective-C* é uma linguagem verbosa (influência do *Smalltalk*) e com muitas potencialidades. Contudo, com o aparecimento de novas linguagens mais modernas, a *Apple* sentiu necessidade de desenvolver uma nova linguagem que fosse mais simples de aprender, surgindo o *Swift*. Essa mudança não trouxe problemas para os desenvolvedores, uma vez que ambas as linguagens são compatíveis e é possível misturá-las num mesmo projeto. Isso acontece porque ambas geram o mesmo código na fase de compilação (Apple Inc., 2010). O *Swift* implementa padrões de programação modernos: as variáveis necessitam de ser inicializadas antes de serem usadas; a memória é gerida de forma automática e permite utilizar *error handlings* que permite recuperar de erros inesperados (Apple Inc., 2017a).

O desenvolvimento nativo para dispositivos *Android* é feito com recurso à linguagem *Java*. O *Android* é desenvolvido por várias empresas, sendo a principal a *Google*. Tanto a *Apple* como a *Google* disponibilizam um conjunto de bibliotecas que facilitam a criação de aplicações robustas e a sua manutenção. Ao facilitarem o processo de desenvolvimento, esperam ter mais desenvolvedores a criar aplicações (Android, n.d.).

O *Apache Cordova* é um projeto *open source*, que permite a criação de aplicações móveis com a utilização de HTML, CSS e *JavaScript*. Oferece várias funcionalidades e tem acesso a vários recursos dos dispositivos, como o acesso aos contactos telefónicos, câmara, microfone e galeria de imagens (Saleh et al., 2016a).

O *React Native* foi desenvolvido e é mantido pelo Facebook e permite desenvolver aplicações com a utilização do *React*. No final, os programas são convertidos em código nativo. O *React Native* obriga a boas práticas de programação, porque é baseado em componentes e certos princípios que no *Cordova* não existem (Saleh, Holmes, Bray, & Yusuf, 2016b). O *React Native* utiliza uma DOM virtual, que permite que apenas seja renderizado o que foi alterado em vez de recarregar toda a *view* de uma aplicação (Saleh, Holmes, Bray, & Yusuf, 2016c).

Tanto as aplicações baseadas em tecnologias *web* como as nativas necessitam de estabelecer ligação com as bases de dados, através de *API's* (*Application Programming Interfaces*). As API implementam a arquitetura *REST* em vez do *SOAP*. A *REST* toma partido dos métodos de pedido HTTP e retorna os resultados ou em formato XML ou *Json*, que é o formato mais utilizado atualmente. A aplicação não pede diretamente informação ao servidor, sendo esta uma tarefa da API. É indispensável proteger a API contra acessos não autorizados e garantir a integridade dos

dados e autenticação dos utilizadores. É importante limitar o número máximo de pedidos por um certo período de tempo para que o servidor possa processar todos os pedidos (Santos, 2016).

A Tabela 3 resume as características das arquiteturas de *software* REST e SOAP.

Arquiteturas	REST	SOAP
Natureza	Estilo de arquitetura de software	Protocolo de comunicação
Dependência de HTTP	Muito dependente	Pouco dependente
Formatos suportados	JSON, YAML, XML	XML
Métodos	Métodos uniformes (GET, POST, UPDATE DELETE)	Muitos métodos customizados
Complexidade	Mais simples e adequada para serviços web.	Bastante superior ao REST
Recursos	Muitos recursos URI's (Uniform Resource Identifier)	Utilização de poucos recursos URI's.
Desempenho	Superior ao SOAP	Inferior ao REST

Tabela 3- Comparação de arquiteturas de *software* REST e SOAP.

Para dar suporte à *App* PT-UA é necessária uma arquitetura REST ou SOAP que permita gerir os utilizadores e adicionar e gerir conteúdos. Para o desenvolvimento dessa arquitetura *e versão Web* existem várias opções, das quais se destacam duas: o Laravel e o CodeIgniter.

O Laravel¹⁷ é uma *framework* PHP muito utilizada, robusta e com mais funcionalidades que o CodeIgniter, como um sistema de comunicação com a base de dados chamado *Eloquent ORM* (*Object Relational Mapper*), que permite abstrair a grande complexidade de criar certos pedidos à base de dados. Tem um sistema que permite gerar as tabelas de bases de dados de forma simples e mais rápida, através das *Migrations* e do *Schema builder*. Possui um sistema de rotas que torna muito fácil gerar URL dinâmicos, um sistema de autenticação embutido muito seguro e fácil de expandir as funcionalidades. Aplica o modelo MVC (Model View Controller), sendo bastante simples de implementar as API e apresenta um maior conjunto de proteções se comparado com o CodeIgniter, como, por exemplo, limita o número de pedidos à API por IP a cada 60 segundos (Laravel, n.d.).

O CodeIgniter¹⁸ é uma *framework* PHP robusta, de reduzidas dimensões (2MB não sobrecarregando o servidor), implementando a arquitetura MVC, tendo uma documentação simples e completa. Apresenta uma boa *performance*, embora tenha um desempenho menor e menos funcionalidades que o Laravel, e possui um conjunto de métodos que facilitam a comunicação com

¹⁷ Consultar www.laravel.com para mais informações sobre a Framework Laravel.

¹⁸ Consultar <https://codeigniter.com/> para saber mais informações sobre a Framework CodeIgniter.

as bases de dados e no desenvolvimento de API. Foi desenvolvida com uma forte componente de segurança, com proteção contra-ataques CSRF e XSS (CodeIgniter, n.d.).

Relativamente às bases de dados, existem bases de dados relacionais e não relacionais. As relacionais são mais vulneráveis a ataques. As não relacionais permitem uma maior escalabilidade e acesso mais rápido, uma vez que não estabelecem relações entre as tabelas da base de dados. A proteção da base de dados deve ser um assunto a tratar atentamente, uma vez que os atacantes podem aceder a informações valiosas e confidenciais (Hou et al., 2016). As bases de dados relacionais estabelecem a relação entre duas ou mais tabelas, facilitando a interpretação dos dados (Hou et al., 2016).

Dentro do modelo de bases de dados relacionais, uma das opções de SGBD (Sistema de Gestão de Base de Dados) mais conhecida e utilizada é o MySQL¹⁹, sendo de código aberto, com desempenho comprovado, confiável e fácil de utilizar, sendo utilizada em muitas páginas *web* e aplicações como o Facebook, Twitter e Youtube (Oracle, n.d.). O SQLite²⁰ é outro tipo de SGBD que utiliza bases de dados relacionais. É muito utilizado por não ser necessário qualquer configuração, e por gravar os seus dados em ficheiros em disco. Não utiliza dependências externas e costuma ser de acesso mais rápido que muitas outras opções de SGBD que fazem uso de bases de dados relacionais, como o caso do MySQL (SQLite, n.d.).

Dentro dos SGBD que recorrem a bases de dados não relacionais, um dos mais conhecidos e utilizado é o MongoDB²¹, sendo este gratuito, de código aberto e que armazena os seus dados sob a forma de JSON em ficheiros armazenados em disco, tratando-se de um sistema mais rápido que o MySQL (MongoDB, 2018). Outro exemplo de SGBD com recurso a bases de dados não relacionais é o caso do Cassandra²², sendo uma excelente opção quando se pretende uma grande escalabilidade sem comprometer o desempenho. Suporta replicações de bases de dados ao longo de vários *datacenters*, reduzindo a latência no seu acesso, e que em caso de falha de um *datacenter*, os restantes podem repor rapidamente os dados perdidos (Apache Cassandra, 2016).

2.3 Plataformas e aplicações de suporte a programas de tutoria

A existência de plataformas e aplicações de apoio a programas de tutoria não é algo novo, existindo algumas alternativas viáveis no mercado. Contudo, estas não permitem ter o grau de personalização que uma plataforma desenvolvida de raiz. Além desta última abordagem permitir uma redução dos

¹⁹ Para mais informações sobre o MySQL, consultar <https://www.oracle.com/br/mysql/index.html>.

²⁰ Para mais informações sobre o SQLite, consultar <https://www.sqlite.org/features.html>.

²¹ Para mais informações sobre o MongoDB, consultar <https://www.mongodb.com/what-is-mongodb>.

²² Para mais informações sobre o Cassandra, consultar <http://cassandra.apache.org/>.

custos, uma vez que é omitido o pagamento de mensalidades a outras empresas para a utilização das plataformas e aplicações.

No mercado conhecem-se alguns softwares de apoio a programas de tutoria, optando-se por analisar o Achiiva, Chronus, Knox, Graduway, MentorNet e Mentor Scout.

O Achiiva²³ é um *software* disponível em iOS e Android, utilizado por mentores e *coaches* no suporte das suas tarefas. Foi desenvolvido pela empresa Achiva PTY LTD, sendo um *software* pago e cujos preços (mensalidades) variam consoante o número de utilizadores. Permite juntar na mesma aplicação os serviços de calendário, e-mail, *chat*, além de permitir estabelecer objetivos para os seus estudantes/clientes. Permite partilhar conteúdos, como fotos, vídeos, pdf e acompanhar o desenvolvimento das tarefas, gerando dados estatísticos, como gráficos (Achiiva, 2016).

O Chronus²⁴ é um *software* de apoio a programas de mentoria, oferecendo uma plataforma configurável e com um algoritmo que faz com que a seleção de um mentor/estudante seja feita com base no Quociente de Inteligência (QI) e perfil de utilizador. É um *software* pago (mensalidades) e cujos preços variam consoante os planos e funcionalidades pretendidas. Permite acompanhar o progresso dos estudantes. Além de um *backoffice* administrativo, oferece aplicação móvel (disponível para iOS e Android) para que possa ser utilizado em qualquer lugar. É um *software* que facilita e incentiva a comunicação entre mentores e tutorandos através de *chat* e partilha de conteúdos. A Bridgestone e a Universidade do Colorado são dois exemplos de organizações que recorrem a este *software* (Chronus, 2018).

A empresa EMentor Connect²⁵ desenvolveu o *software* Knox de apoio a programas de tutoria. Adapta-se a necessidades específicas de cada programa, simples de utilizar e facilita todo o trabalho de gestão. Permite o *live chat* e está preparado para dispositivos móveis. É um *software* pago, cujo valor varia consoante as necessidades (eMentorConnect, n.d.).

A empresa Graduway²⁶ desenvolveu um *software* com o mesmo nome, que é utilizado para construir uma comunidade onde mentores e tutorandos podem interagir e partilhar conhecimento. Embora esteja presente em todos os continentes, a sua presença é mais notada nos Estados Unidos e Europa. Em Portugal é usado pela AESE Business School e a Porto Business School. É um *software* pago, mas o seu valor varia consoante o projeto, não havendo no seu *website* qualquer tabela de preços (Graduway, 2017).

²³ Para mais informações sobre o Achiiva, consultar <https://www.achiiva.com/>.

²⁴ Para mais informações sobre o Chronus, consultar <https://chronus.com/>.

²⁵ Para mais informações sobre o eMentorConnect, consultar <http://ementorconnect.com/solution/knox-platform/>.

²⁶ Para mais informações sobre o Graduway, consultar <https://graduway.com/>.

O *software MentorNet*²⁷ é utilizado pela Universidade de Liverpool e Oxford para auxiliar os seus programas de mentoria. Os mentores e estudantes têm o seu próprio perfil e podem comunicar entre si. Permite a troca segura de documentos (pdf, fotos, vídeos) entre mentores e tutorandos. Possui mecanismos que gerem relatórios de desenvolvimento e que permitem acompanhar a sua evolução. Permite a realização de inquéritos para avaliar o valor dos programas de mentoria. Foi desenvolvido por uma organização sem fins lucrativos chamada *Great Minds in STEM*, pelo que a sua utilização é gratuita (MentorNet, n.d.).

O *software Mentor Scout*²⁸ permite auxiliar os programas de mentoria através de ferramentas que facilitam a comunicação e ferramentas que geram gráficos que permitem acompanhar as alterações. Os mentores e tutorandos têm o seu próprio perfil e podem comunicar de forma segura. Os mentores podem lançar desafios aos estudantes. É um *software* pago, mas o seu valor varia consoante o projeto, não havendo no seu *website* qualquer tabela de preços (Mentor Scout, 2018).

Alguns autores como Ferdousi e Bari (2015), Moura e Barret-Greenly (2015), Gómez-Doming e Badia-Garganté (2016) e Zydney e Warner (2016), como citado em Griol, Molina e Callejas (2017) referem que alguns estudos recentes apontam para as grandes vantagens e potencialidades da utilização de aplicações móveis no ensino, tornando-o mais interativo. Klemmer (2015) refere que quando se aprende através de dispositivos móveis, o *design* e usabilidade são indispensáveis ao sucesso.

Griol et al. (Griol et al., 2017) desenvolveram uma aplicação multimodal para a universidade onde estudavam, possibilitando o seu uso através do toque, voz ou ambos. Para esse desenvolvimento foi criado um servidor que recebia e transmitia dados para a aplicação, utilizando uma API e uma base de dados desenvolvida em MySQL. A API retornava uma *array* de Json e a aplicação móvel utilizava uma base de dados SQLite para armazenar dados locais. Para utilizar a aplicação é necessário o registo e as funcionalidades que esta oferece dependem do tipo de utilizador e das suas permissões (conforme é estudante ou professor). Oferece vários modos, desde perguntas educativas separadas por categorias e temas; à consulta de livros da Biblioteca (efetuar/renovar reservas); permite consultar informações sobre as localizações dos vários departamentos e pontos de interesse na universidade.

O Unibly²⁹ (Figura 6) é uma plataforma de *mentoring* que disponibiliza também aplicações *móveis* onde os utilizadores colocam os dados de acesso à comunidade a que pertencem e posteriormente fazem *login* via *Facebook* ou com as suas credenciais, ou criam conta. Na criação de conta é

²⁷ Para mais informações sobre a MentorNet, consultar <http://mentornet.org/organization/programs.html>.

²⁸ Para mais informações sobre a Mentor Scout, consultar <https://www.mentorscout.com/>.

²⁹ Para mais informações sobre o Unibly, consultar <http://www.unibly.com/>.

pedido para seleccionar o seu papel (mentor ou mentorando) e dados como nome, género, país, ano de nascimento, experiência profissional e o tipo de ajuda que desejam receber ou oferecer, além de uma breve descrição pessoal. Ao entrar com a sua conta, o utilizador tem acesso a um *feed* de notícias com os últimos *posts* de outros utilizadores, acesso a listas com mentores e mentorandos onde é possível estabelecer ligações para que possa haver um contacto via *chat*. O acesso administrativo para gerir toda a comunidade é feito através do seu *website*, entrando na página administrativa. Na parte administrativa é possível consultar a lista de utilizadores e alterar dados dos utilizadores, ver, editar e apagar notícias e ver notificações. Para aceder ao setor administrativo é necessário ter privilégios administrativos. É utilizado em várias instituições como a *Nanyang Business School* e a *Singapore Poly Business School* (Admin Unibly, 2017; Unibly, 2015).

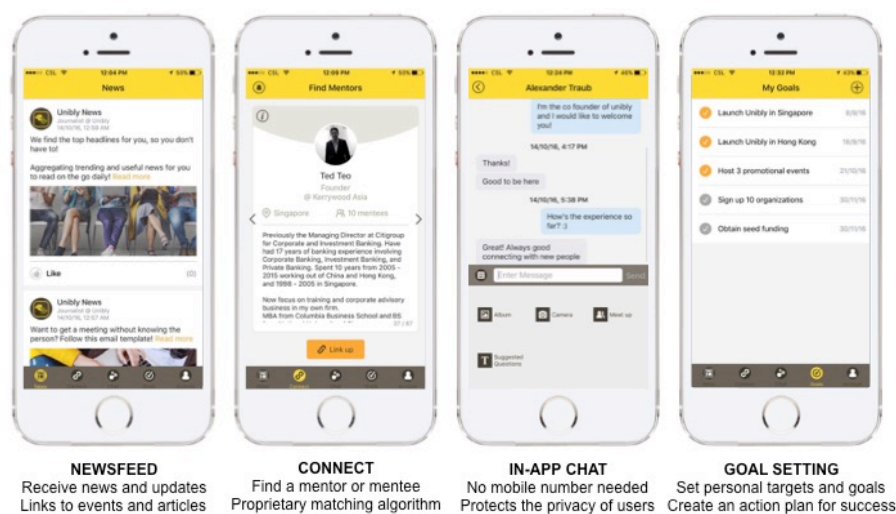


Figura 6 - Alguns ecrãs da plataforma Unibly (Unibly, n.d.)³⁰.

A empresa Mentored³¹ (Figura 7) desenvolveu uma plataforma *online* de mentoria com o mesmo nome, que oferece soluções para grupos de trabalho, empresas e escolas. Esta plataforma permite chamadas de vídeo privado e várias soluções de chat que podem ser customizadas ao gosto dos utilizadores. Pode ser acedida em computador ou em dispositivos móveis permitindo aprender e trocar experiências em qualquer lugar. Oferece um ambiente interativo para que mentores e mentorandos possam interagir, além de fornecerem um conjunto de CMS (*Content Management System*) e análise de dados que facilitam a gestão (Mentored, 2016).

³⁰ Imagem retirada do *website* <http://unibly.squarespace.com/mentoringplatform> em 19/01/2018.

³¹ Para saber mais informações sobre o Mentored, consultar www.mentored.com.

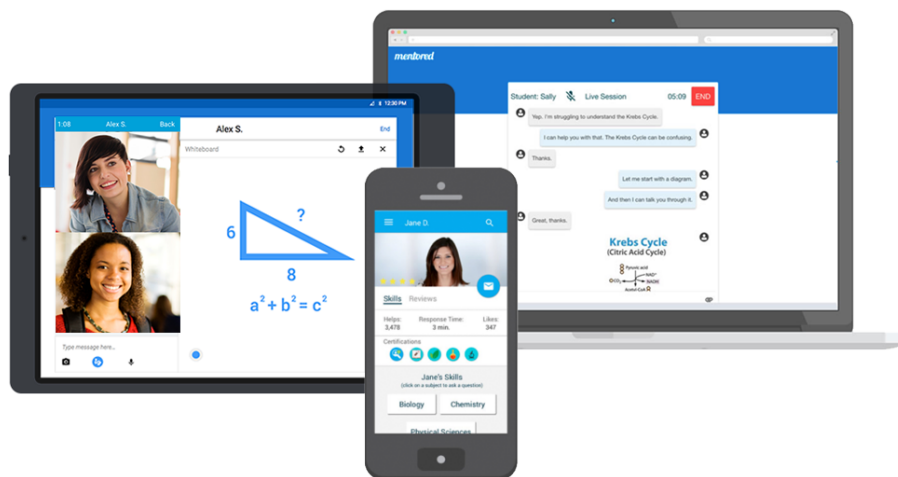


Figura 7 - Alguns ecrãs da plataforma Mentored (Mentored.com, n.d.)³².

A empresa Edmodo³³ (Figura 8) desenvolveu uma plataforma *online* com o mesmo nome, que disponibiliza também aplicações móveis onde professores e estudantes podem conectar-se, partilhar e aprender dentro de uma rede social. Ao entrar na aplicação, os utilizadores deparam-se com cinco separadores inferiores que permitem navegar entre as disciplinas, tarefas, mensagens, notificações e menu. No separador das disciplinas, os estudantes seleccionam uma disciplina e acedem a um conjunto novo de separadores no canto superior que permitem navegar rapidamente entre o *feed* de notícias, ver os membros, recursos disponibilizados e grupos mais pequenos dentro da disciplina. Os encarregados de educação podem acompanhar, tendo acesso aos conteúdos dos seus educandos. No modo de *chat*, os estudantes podem interagir com outros estudantes e professores, para tirar dúvidas (Edmodo, 2018; Mr. Miranda The Tech Guru, 2017).

³² Imagem retirada do website <https://www.mentored.com/platform.html> em 19/01/2018.

³³ Para mais informações sobre o Edmodo, consultar <https://www.edmodo.com/about>.

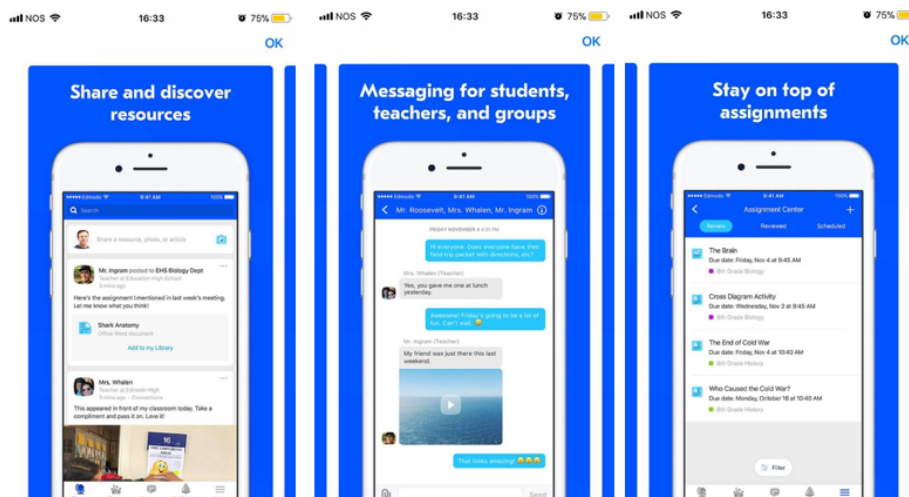


Figura 8 - Alguns ecrãs da plataforma Edmodo (Edmodo, 2018).

A Universidade de Ashford desenvolveu uma aplicação móvel (Figura 9) que permite que os estudantes possam aceder aos conteúdos do seu curso, saber novidades, *posts* de outros utilizadores, cursos complementares com competências transversais, consultar resultados escolares e entregar tarefas das várias UC (Ashford University, 2012).

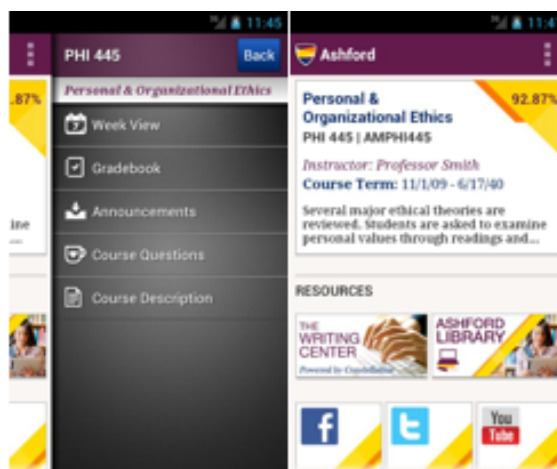


Figura 9 - Alguns ecrãs da aplicação da Universidade de Ashford (Education, n.d.)³⁴.

A Universidade do Norte de Ilinoís disponibiliza uma aplicação móvel que permite saber quando e onde irão ocorrer eventos académicos, saber as principais informações sobre o curso que o estudante frequenta, além de um modo de mapa do *campus* com *GPS* que permite guiar os estudantes, para que estes não se percam (Northern Illinois University, 2011).

³⁴ Imagem retirada do website <https://play.google.com/store/apps/details?id=edu.ashford.talon> em 19/01/2018.

A Universidade do Estado da Carolina do Norte e a Universidade do Oregon utilizaram a plataforma *Guidebook EDU* para desenvolverem as suas aplicações móveis, onde são disponibilizadas informações sobre eventos a ocorrer na universidade, mapa com *GPS* para guiar os estudantes, acesso aos contactos dos professores e outros elementos da comunidade educativa, e acesso a conteúdos relacionados das várias UC (Guide Book Edu, n.d.).

2.4 Considerações finais

A evolução da *Internet* e da *web* tem sido marcada pelo aparecimento de muitas tecnologias digitais e serviços, que impulsionam o aparecimento e desenvolvimento de várias linguagens de programação, tendo sido introduzidas novas funcionalidades. Existe uma dependência cada vez maior da *Internet* e dos websites de redes sociais, passando estes a serem uma forma de comunicação interessante. De forma a potenciar maior aceitação e envolvimento de estudantes e professores, a App PT-UA deverá incluir funcionalidades e características usualmente disponíveis em *websites* de redes sociais. Nomeadamente, página de perfil de utilizador onde seja possível este modificar os seus dados e imagem de perfil; integração de comunidades e grupos; possibilidade de *chat* com outros utilizadores; capacidade para criar *posts* e expressar a sua opinião; criação e divulgação de eventos, formações, etc.

A realização de testes de usabilidade durante o desenvolvimento de uma App permite encontrar e solucionar, de forma fácil, problemas, e solucioná-los antes da mesma ser disponibilizada aos seus utilizadores. O princípio de não repetir código e a utilização de *Dependency Injection* é igualmente determinante para que a construção de uma aplicação seja o mais eficiente e fluente possível.

No desenvolvimento da App PT-UA será seguido o princípio da POO, e os bons princípios de *design* de software apresentados neste capítulo. Pretende-se que o desenvolvimento seja o mais eficiente possível e que permita que a mesma seja escalável para muitos utilizadores, sem que seja necessário refazer código.

A Tabela 4 sintetiza e compara as plataformas de apoio a programas de tutoria abordadas, de forma a facilitar a sua compreensão. Para a elaboração da Tabela 4 foi feita uma consulta aos *websites* das plataformas, e feita uma pesquisa na *Apple Store* e na *Google Store*.

Plataformas	Achiiva	Chronus	Ementor Connect	Graduaway	Mentor Net	Mentor Scout	Unibly	Mentored	Edmodo
Características									
Plataforma de gestão (backoffice)	x	x	x	x	x	x	x	x	x
versão móvel (ios e android)	x	x	x	x	x	x	x	x	x
Algoritmo de seleção mentor / aluno		x							
Chat	x	x	x	x	x	x	x	x	x
Partilha de conteúdos	x	x	x	x	x	x	x	x	x
Mensalidades	x	x	x	x		x	x	x	x
Grande capacidade de personalização		x	x						
Página de perfil	x	x	x	x	x	x	x	x	x

Tabela 4 - Comparação de várias plataformas de apoio a programas de tutoria.

Após a análise das várias plataformas e aplicações, e da sua comparação, foram identificadas algumas funcionalidades interessantes para incorporar no PT-UA, nomeadamente, um painel administrativo onde os tutores e mentores possam adicionar, editar e apagar conteúdos; o modo de *GPS* para permitir que os novos estudantes da UA não se sintam perdidos dentro dos *campi*; o modo de *chat* para facilitar a comunicação entre tutores, mentores e tutorandos; um modo onde seja fácil conhecer quais os eventos a decorrer e os cursos disponíveis sobre competências transversais.

O desenvolvimento de uma plataforma e aplicação própria é uma opção que permite um elevado grau de personalização e uma redução considerável de custos, não sendo necessário pagar mensalidades pela utilização de plataformas disponibilizadas por terceiros. Em contrapartida, obriga a que seja necessário desenvolver todo o código e funcionalidades.

No próximo capítulo apresenta-se o procedimento metodológico que orienta este trabalho de investigação.

3. Metodologia

Neste capítulo é realizada uma apresentação dos objetivos de investigação definidos neste trabalho de investigação, assim como do procedimento metodológico aplicado.

3.1 *Objetivos e abordagem de investigação*

Este trabalho tem como objetivos de investigação o desenho e implementação de uma aplicação de suporte ao PT-UA, assim como uma versão *Web* que permita facilitar a gestão de conteúdos e utilizadores. Pretende-se potencializar a aproximação com o seu público-alvo através da integração de novas tecnologias em junção com as redes sociais.

A App PT-UA tem como objetivo melhorar o processo de comunicação entre todos os participantes do programa, através da comodidade de aceder às informações a partir do seu *smartphone* e da possibilidade de participar em conversas em tempo real. Pretende-se também que facilite a navegação dos novos estudantes dentro dos *campi*, através da disponibilização de uma funcionalidade de navegação GPS dentro da App. Tem ainda como objetivo oferecer comodidade aos seus utilizadores, através da facilidade em obter as informações sobre os principais eventos a decorrer na universidade. A disponibilização da funcionalidade de colocação de publicações pelos seus membros, estimulando a criação de uma comunidade *online* e a melhoria dos laços relacionais, é outro dos objetivos.

Um trabalho de investigação como este é um trabalho árduo, composto por vários avanços e recuos, mantendo o foco na descoberta das melhores formas de conceptualizar e implementar a App PT-UA. Assim, a abordagem metodológica que orienta o objetivo principal deste trabalho é a Investigação de Desenvolvimento (Coutinho & Chaves, 2001). A metodologia de Investigação de Desenvolvimento segue as seguintes fases: análise de problemas práticos, desenvolvimento da solução, e avaliação e teste no terreno. Na fase final, recomenda-se uma reflexão sobre eventuais investigações futuras e a elaboração da documentação de todo o processo. Dentro desta aplica-se a forma de desenvolvimento de objeto. Neste caso, o nosso objeto é referente à App PT-UA.

Assim, esta dissertação passa por várias fases, tal como representado na Figura 10, sendo que numa primeira fase é feito o levantamento do estado de arte, que engloba a procura e análise de Apps similares, assim como boas práticas de desenvolvimento de software, e formas de desenvolver Apps.

Na segunda fase é feita a implementação dos *layouts* em código, oferecendo já as funcionalidades principais, como a navegação *GPS* para facilitar a navegação dos estudantes dentro dos *campi*; a possibilidade de aceder aos principais eventos e adicionar eventos; a capacidade de dar *like*, de comentar um determinado *post* e criar novos *posts*; ou a possibilidade de *chat* em tempo real e de

editar a sua página de perfil. Para a implementação dos *layouts* é utilizado o *react native* para a componente gráfica (*UI*) da *App*, e o *redux* para a gestão de estados da *App*. Para o *backend* é utilizado *laravel*, e o *pusher* para as conexões em tempo real do *chat*. O modo de navegação *GPS* utiliza um módulo externo que é utilizado em conjunto com o *react native* e que permite aceder à localização do utilizador e apresentar rotas, consoante a escolha de um ponto de destino.

Na terceira fase são realizados testes de usabilidade a um grupo de participantes do programa de forma a conhecer alguns dos problemas de usabilidade que possam surgir.

Na quarta fase ocorre a apresentação dos resultados, e comporta a análise dos resultados e sugestões de melhoria. As sugestões de melhoria obtidas de testes de usabilidade são bastante úteis para que possam ser corrigidas falhas antes do lançamento final, evitando custos desnecessários mais tarde. Após esta fase é iniciada a correção e implementação de algumas das sugestões de melhoria resultantes dos testes de usabilidade, correspondendo a uma segunda fase de implementação.

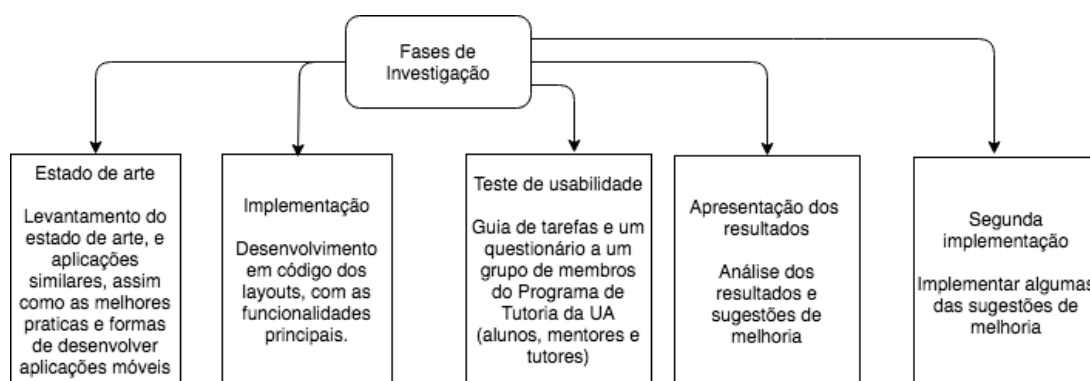


Figura 10 - Fases de investigação.

Na metodologia empírica tradicional, o período de experiência dura em média 60 minutos, e na metodologia de Investigação de Desenvolvimento este pode demorar dias, semanas, meses, sendo necessário muitas vezes a cooperação de investigadores, professores, conceção de protótipos, avaliação e melhorias (Coutinho & Chaves, 2001). Nesta dissertação, o período de experimento dura vários meses, tendo em conta que necessita da cooperação de investigadores, professores, mentores e estudantes do 1.º ano, além da conceção de protótipos, avaliação e melhorias a realizar consoante os resultados dos testes de usabilidade. O desenvolvimento de um protótipo funcional como o apresentado nesta dissertação é um processo longo, com muitos avanços e recuos, seja no desenvolvimento do *design* seja na implementação de todo o código e sua lógica associada. Ou ainda, o processo do desenvolvimento do guião e questionário na realização dos testes de usabilidade, assim como o da análise dos resultados e suas correções na fase de implementação das sugestões de melhoria.

3.2 Conceção dos layouts

Nesta dissertação optou-se por não realizar um levantamento de necessidades junto dos tutores, mentores e tutorandos através do recurso a entrevistas, e em contrapartida realizar um *Benchmarking* para encontrar quais as funcionalidades mais utilizadas em aplicações similares às que se pretende implementar. Essas funcionalidades mais utilizadas em aplicações similares, e que permitiram definir as funcionalidades a implementar na *App* PT-UA, foram aprofundadas no capítulo dois. A concepção dos *layouts* corresponde à segunda fase da metodologia, ou seja, pertence à fase de implementação.

Apesar de ser aconselhado o desenvolvimento de *mockups* antes da implementação em código, devido à equipa de desenvolvimento ser apenas constituída por uma pessoa e por motivos de tempo, opta-se por realizar o *design* já em código. Assim, a implementação tem em conta as recomendações apresentadas anteriormente, de forma a garantir uma interface de utilização agradável e adequada aos utilizadores. Os princípios de desenvolvimento apresentados anteriormente também são essenciais para que o código desenvolvido seja o mais reutilizável, simplificado e mais fácil de manter no futuro, sendo uma base interessante para futuros aperfeiçoamentos.

O desenvolvimento já em código permite que os testes de usabilidade sejam mais reais do que através do recurso a *mockups* dinâmicos, uma vez que é possível utilizar os recursos, como o de navegação *GPS*. A *App* PT-UA contém sete ecrãs principais. São estes:

- Um ecrã inicial, onde o utilizador escolhe entre entrar ou registar;
- Um ecrã de registo;
- Um ecrã de entrar;
- Um ecrã de comunidade;
- Um ecrã de eventos;
- Um ecrã de navegação *GPS*;
- Um ecrã de perfil;

O ecrã inicial, quando o utilizador não tem o *login* feito, serve para este escolher entre a opção de registar conta ou entrar com a sua conta. Quando o utilizador já está com *login* feito, a página inicial passa a ser a de comunidade, onde este poderá consultar publicações de outros utilizadores, fazer *like* nas publicações, comentar e criar publicações. Pretende permitir um contacto mais frequente entre todos os participantes, além de se tirar partido da utilização dos *smartphones* e da *Internet*, através de um modo cómodo. O ecrã de eventos visa permitir consultar os eventos que decorrem nos *campi*, além de permitir adicionar eventos caso os utilizadores tenham permissão para adicionar ou editar os eventos. Será assim mais fácil para os estudantes, e outros participantes

no PT-UA, acederem às informações dos eventos. No ecrã de navegação *GPS* pretende-se que permita aos utilizadores obterem a sua localização no mapa, e selecionarem um local para onde desejam ir, para que lhe seja apresentado um percurso até ao local desejado. Tem assim a função de ajudar os novos estudantes nos primeiros tempos dentro dos *campi*, sem ser necessário estes pedirem informações a terceiros. O ecrã de perfil visa permitir aos utilizadores acederem ao seu perfil e visualizar e editar os seus dados, como nome, e-mail, e foto de perfil. Através do ecrã de *chat* pretende-se assistir os utilizadores em comunicações em tempo real, e no desenvolvimento de laços relacionais mais fortes. Esse ecrã deve apresentar a imagem e nome do utilizador para cada inserção de texto no *chat*, de forma a facilitar o reconhecimento do autor.

Relativamente ao código desenvolvido, este é disponibilizado em apêndice, podendo ser descarregado a partir do *Github*, após ser autorizada a sua visualização. O *Github* é um dos repositórios mais utilizados do mundo, e permite fazer um controlo de versões, assim como disponibilizar o seu acesso a toda uma comunidade, no caso dos repositórios públicos. Nos repositórios privados, o seu acesso só é feito mediante aprovação por parte dos administradores desses repositórios. Foram utilizados dois repositórios privados do *Github*, um para a *Api* e versão *web*, e o outro contém o código referente à *App*. Optou-se por não colocar público por questões de confidencialidade.

3.3 Recolha e análise de dados

Usualmente, a recolha de dados no âmbito da metodologia de Investigação de Desenvolvimento pode suportar-se quer em ferramentas quantitativas quer qualitativas (Coutinho & Chaves, 2001). Métodos qualitativos baseiam-se na observação do comportamento dos utilizadores e a análise dos dados requer uma análise do contexto em que estes respondem. Métodos quantitativos baseiam-se em dados recolhidos indiretamente, como saber onde os utilizadores clicaram, que ecrãs visitaram e número de cliques para completar uma tarefa (Teixeira, 2017).

A recolha de dados no âmbito da fase de testes de usabilidade é bastante importante, porque permite saber, nomeadamente, o grau de satisfação e estado de espírito dos utilizadores durante a utilização da *App* PT-UA, assim como encontrar falhas na usabilidade. A correção dessas falhas permitirá posteriormente melhorar a experiência de utilização. Os testes de usabilidade permitem avaliar o UX (*User eXperience*) e obter uma ideia do grau de satisfação e da experiência de que a sua utilização proporciona. Estes testes permitem encontrar falhas que mais tarde poderiam comprometer o sucesso da aplicação. Antes de escolhermos os métodos a utilizar é necessário analisar de forma cuidada alguns dos maiores especialistas em UX, e os métodos utilizados pelos mesmos, de forma a ser mais fácil aplicar testes que sejam adequados a esta *App* PT-UA. Segundo Shedroff (n.d.), UX é a experiência entre um utilizador e uma aplicação e a sua facilidade de

utilização (Allaboutux.org, n.d.-g). Refere-se, portanto, à percepção da forma como as pessoas interagem, percebem e se sentem com a utilização de um produto. Para se obter uma ótima experiência de utilização é necessário satisfazer de forma eficaz as verdadeiras necessidades dos utilizadores (Norman & Nielsen, 2018). Para oferecer uma ótima experiência de utilização, uma *app* necessita de ir ao encontro das verdadeiras necessidades dos utilizadores, procurando deixar os utilizadores satisfeitos. Por vezes, as aplicações são um pouco confusas levando a que os utilizadores se sintam confusos e frustrados na sua utilização, pelo que quando uma *app* oferece uma interface simples já provoca um sentimento de satisfação. Uma interface deve ser pensada nos seus utilizadores finais, focando-se em ser simples, intuitiva e o mais adequada possível às necessidades dos seus utilizadores.

A utilização do UX no processo de desenvolvimento traz um conjunto de vantagens, como o incremento de produtividade; aumento das vendas; redução de custos de formação e suporte, redução do tempo de desenvolvimento e aumento da satisfação dos consumidores (User Experience Professionals Association, n.d.), tendo em conta que permite o desenvolvimento de interfaces mais simples e adequadas aos utilizadores. Uma interface mais simples ajuda a incrementos de produtividade se comparado com interfaces mais complexas. No caso do PT-UA, o UX irá auxiliar o processo de desenvolvimento de uma *app* simples e agradável de utilizar, permitindo uma maior produtividade por parte dos tutores e mentores, assim como na redução da curva de aprendizagem que os utilizadores necessitam para utilizá-la.

Um conceito muito importante para avaliar o UX é o conceito de usabilidade. Segundo Nielsen (2012), usabilidade é a facilidade com que os utilizadores utilizam a aplicação; o tempo que estes demoram a realizar uma tarefa após aprenderem; a capacidade de memorização de um utilizador experiente quando deixa de utilizar a aplicação por um longo período de tempo e o tempo que demora a recuperar a proficiência; o número de erros que estes cometeram; se conseguem recuperar do erro e o grau de satisfação na utilização. Os testes podem ser feitos em laboratórios ou numa pequena sala, o importante é simular as condições reais de utilização da aplicação.

Para a avaliação UX existe um conjunto de métodos, dos quais se deve escolher um que se adeque ao tema a ser avaliado. Os métodos de estudo de campo geralmente são de curta duração, embora possam ser de longa duração. Estes têm a vantagem de serem em contexto real: métodos de estudo de laboratório que são realizados em locais físicos fixos (laboratórios), sendo os contextos de uso simulados; métodos de estudo *online* que são realizados a utilizadores *online*, podendo estes ser anónimos ou apenas a pessoas que receberem convite e métodos de questionários que utilizam escalas para avaliar certos aspetos de elementos UX (Allaboutux.org, n.d.-e). Estes estudos podem ser feitos em várias fases de desenvolvimento, desde os testes de conceito; primeiros protótipos; protótipos funcionais; ou a produtos já lançados no mercado. Uma das variáveis a estudar passa por

determinar quem irá fazer os testes; se peritos em UX, se um utilizador de cada vez; se um grupo de utilizadores, se um conjunto de pares de utilizadores (Allaboutux.org, n.d.-e).

Existem vários métodos famosos como o AXE, *Emocards*, SAM, *Emofaces*, 3E, Testes Exploratórios e MAX. O AXE é um método qualitativo que recorre a utilizadores singulares e tem como base a utilização de entrevistas. É um método de avaliação e uma forma de obter sugestões de melhoria (Allaboutux.org, n.d.-b). O *Emocard* consiste num conjunto de cartões através dos quais os utilizadores expressam as suas emoções sobre determinadas partes da aplicação sujeitas a avaliação. É rápido e simples de fazer, não requer formação, é económico e pode ser facilmente integrado em qualquer estudo (Allaboutux.org, n.d.-c). O *Self Assessment Scale* (SAM) permite saber o grau de satisfação dos utilizadores; se os utilizadores se sentem com controlo sobre a aplicação e se estão calmos ou excitados com a interface (Allaboutux.org, n.d.-f). O método de *Emofaces* consiste na utilização de cartões com várias expressões faciais e é útil quando a natureza do produto dificulta de certa forma a verbalização. Este método pode ser usado em questionários na Internet ou entrevistas formais (Allaboutux.org, n.d.-c). O método 3E (*Expressing Experiences and Emotions*) pode ser utilizado em estudos de campo e permite conhecer o estado de espírito do utilizador durante a utilização da aplicação (Allaboutux.org, n.d.-a). Os testes exploratórios geralmente apresentam um protótipo a um utilizador, solicitam-lhe a execução de um conjunto de tarefas e é pedido que refiram outros produtos/aplicações similares que tenham utilizado ou tenham conhecimento (Allaboutux.org, n.d.-d). O método MAX (*Method of Assessment Manikin*) utiliza cartões com avatares e geralmente é aplicado após a utilização dos *mockups*, protótipos. Aborda quatro categorias: como o utilizador se sentiu a utilizar a aplicação; se considerou a utilização fácil; se teve utilidade e qual o desejo de voltar a utilizá-la (Allaboutux.org, n.d.-f).

As escalas de Likert são utilizadas nos questionários devido a serem escalas fáceis de construir, além de permitirem saber a opinião dos entrevistados sobre questões por vezes complexas, contudo os entrevistados tem tendência a concordar com as questões que lhes são colocadas (Llauradó, 2015).

Para a avaliação do protótipo da *App* PT-UA recorre-se a testes exploratórios onde é pedido aos utilizadores que realizem um conjunto de tarefas contidas no guião apresentado na Tabela 5.

Guião
Registe uma nova conta de utilizador.
Faça login.
Aceda ao ecrã de perfil e adicione uma foto de perfil e uma breve descrição.
Consulte o ecrã da Comunidade
Faça gosto numa notícia à sua escolha e escreva um comentário.
Crie uma nova notícia.
Aceda ao calendário de eventos.
Adicione um novo evento no calendário.
Aceda ao chat e inicie uma conversa.
Aceda ao Mapa, obtenha a sua localização e peça indicações para um dos departamentos.
Faça log-out.

Tabela 5 – Guião fornecido aos participantes do estudo.

Antes de ser solicitada a realização das tarefas fez-se uma breve apresentação para que os utilizadores compreendessem o que estava a ser pedido. Enquanto os utilizadores realizavam as tarefas (um utilizador de cada vez), um observador registou e colocou perguntas pertinentes para compreender quais as dificuldades que estes encontravam e o seu estado de espírito. Seguidamente foi pedido aos utilizadores que respondessem a um breve questionário, que utilizou o método de *Emocards*, Escala de Likert e questões abertas. A combinação destes métodos permite ter uma ideia aproximada do grau de satisfação dos utilizadores e as principais dificuldades encontradas por estes, para que eventuais problemas possam ser resolvidos antes da implementação final. As questões com recurso ao método *Emocards* estavam relacionadas com as várias funcionalidades da aplicação e a avaliação se estas funcionalidades eram úteis para os utilizadores. A Escala de Likert ajudou a avaliar o grau de dificuldade de cada um dos ecrãs, e as questões abertas permitiram conhecer quais as funcionalidades que os utilizadores consideraram mais importantes, assim como as que estes menos gostaram e as sugestões de melhoria.

O estudo foi feito com cinco potenciais utilizadores (quatro mentores, um tutorando e um tutor) participantes atualmente no PT-UA e realizado no Departamento de Comunicação e Arte da Universidade de Aveiro. O teste foi efetuado numa das salas de estudo do departamento, com recurso a um portátil que utilizou o EXPO para compilar o código e renderizar num *smartphone*, tornando possível a realização dos testes. O EXPO é um software gratuito desenvolvido por uma comunidade de desenvolvedores, muitos deles antigos funcionários e desenvolvedores do *React*. Este permite compilar o código do *react native* para as linguagens nativas dos *smartphones* e renderizar as *app* nos mesmos, sem ser necessário estas estarem na *App Store* ou *Google Play*. É muito útil na fase de teste e no desenvolvimento, por conter um conjunto de funcionalidades que reduzem a complexidade de código que é necessário para comunicar com componentes dos

smartphones, como é o caso da geolocalização e biblioteca de imagens dos dispositivos. Após todo o código estar desenvolvido, esta permite compilar o código todo de forma otimizada, facilitando o processo de lançamento na *App Store* ou *Google Play*. Na Tabela 6 são apresentadas as questões que foram colocadas aos participantes durante o teste.

Número da questão	Questão
1	Cargo: aluno <input type="checkbox"/> mentor <input type="checkbox"/> tutor <input type="checkbox"/>
2	Departamento:
3	Idade:
4	Qual o grau de satisfação geral na utilização da Aplicação? (EMOCARDS)
5	Qual o grau de satisfação relativamente à navegação (menu inferior e lateral)? (EMOCARDS)
6	Qual o grau de dificuldade relativamente à navegação (menu inferior e lateral)? (Escala de Likert)
7	Qual o grau de dificuldade relativamente à criação de uma nova conta? (Escala de Likert)
8	Qual o grau de dificuldade relativamente à criação de uma nova notícia? (Escala de Likert)
9	Qual o grau de satisfação relativamente ao feed de notícias e na possibilidade de adicionar novos conteúdos, comentar e dar "gosto"? (EMOCARDS)
10	Qual o grau de satisfação relativamente ao ecrã de perfil e edição dos seus dados de utilizador? (EMOCARDS)
11	Qual o grau de dificuldade relativamente à edição dos seus dados de perfil? (Escala de Likert)
12	Qual o grau de satisfação relativamente ao ecrã de chat? (EMOCARDS)
13	Qual o grau de dificuldade relativamente à utilização do chat? (Escala de Likert)
14	Qual o grau de satisfação relativamente ao ecrã de Mapa Navegação? (EMOCARDS)
15	Qual o grau de dificuldade relativamente à utilização do mapa de navegação? (Escala de Likert)
16	Qual o grau de dificuldade relativamente à criação de um novo evento? (Escala de Likert)
17	Qual o grau de satisfação relativamente ao ecrã de eventos e na possibilidade de adicionar novos conteúdos? (EMOCARDS)
18	Que funcionalidades achavas importante adicionar? (Questão aberta)
19	Quais os aspetos da Aplicação que menos gostaste? Mudarias algo? (Questão aberta)

Tabela 6 – Questões colocadas aos participantes do teste de usabilidade

A elaboração deste questionário teve como base os autores conceituados do UX e as suas recomendações, de forma a que fosse possível tirar conclusões úteis para a análise de resultados.

A salientar que na escolha dos participantes do estudo atendeu-se a que a amostra fosse constituída por elementos com cargos diferentes, departamentos diferentes e sexo diferente. Na Figura 11 é apresentada a distribuição dos participantes do estudo pelas Unidades Orgânicas da UA. É importante referir que para que esta recolha de dados fosse possível foi necessário pedir autorização à equipa reitoral da UA bem como o acesso aos dados dos participantes do PT-UA. Antes da realização da recolha dos dados e testes de usabilidade foi obtido dos participantes o consentimento informado, de forma a cumprir os requisitos legais. Foi também entregue o guião com um conjunto de tarefas que estes teriam de desempenhar na realização dos testes, e o formulário com o conjunto de questões que permitiram avaliar a usabilidade e dificuldade de utilização dos vários ecrãs da App PT-UA.

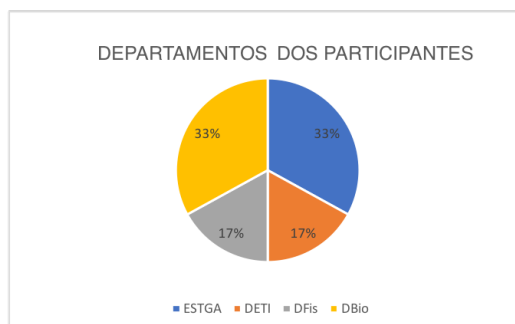


Figura 11 – Distribuição dos participantes por Unidade Orgânica.

Sobre a amostra usada na recolha de dados para o teste de usabilidade ainda a referir que 66% dos participantes tinham o cargo de Tutor, 17% o cargo de Mentor, e 16% o cargo de Tutorando, como pode ser visualizado na Tabela 7.

Cargo	Frequência absoluta	Frequência relativa
Estudante (primeiro ano)	1	16%
Mentor	1	17%
Tutor	4	67%

Tabela 7 – Distribuição dos participantes por cargo no PT-UA.

A idade média dos participantes era de 26 anos, sendo que as pessoas mais novas tinham 19 anos, e a mais velha tinha 53 anos. Dos participantes, 67% eram do sexo feminino e 33% do sexo masculino, provenientes de diferentes unidades orgânicas, cursos e com diferentes necessidades específicas de forma a assegurar a obtenção de resultados mais interessantes.

Todos os participantes foram voluntários, e surgiram da sua demonstração de interesse em participar neste estudo, após a leitura de um e-mail reencaminhado a todos os participantes do PT-UA.

4. Resultados

Neste capítulo é apresentado inicialmente qual o contributo esperado da *App* PT-UA para o Programa de Tutoria da Universidade de Aveiro, sendo posteriormente explicado o processo de implementação das funcionalidades e apresentado um mapa da navegação do utilizador na *App*. Seguidamente são apresentados os resultados obtidos nos testes de usabilidade efetuados, assim como as sugestões de melhoria identificadas, e quais as melhorias implementadas com base nas mesmas.

4.1 Contributo da App PT-UA

O PT-UA surgiu com o objetivo de promover o sucesso escolar e prevenir o abandono, tendo como base a sua inspiração em anteriores programas de tutoria, que foram sucessivamente implementados na UA, bem como em outras iniciativas em funcionamento nacional e internacionalmente. Visa o acompanhamento dos novos estudantes na transição para o ensino superior, na promoção do bem-estar pessoal, na aquisição de competências transversais, e na sua integração junto da restante comunidade académica. Embora o foco do PT-UA seja os estudantes do primeiro ano, outros estudantes de anos posteriores, ou estudantes internacionais e de Erasmus podem igualmente usufruir de apoio. O programa prevê três tipos de papéis, o papel de tutor, o papel de mentor e o papel de tutorando (estudante tutorado). O modelo adotado prevê que cada tutor tenha ao seu encargo três mentores para o auxiliarem no processo, e que cada mentor tenha ao seu encargo cinco tutorandos. Este modelo é ajustado às necessidades de cada contexto. A Figura 12, retirada da página de facebook do PT-UA, sintetiza os valores principais do programa.



Figura 12 – Principais valores do PT-UA (Programa de Tutoria / PT-UA, 2018)

A abordagem inicial ocorre na primeira semana de aulas, no período ainda do acolhimento, onde o programa procura informar os novos estudantes da existência do mesmo e das suas vantagens, nomeadamente, no acompanhamento e integração dos estudantes no meio académico, nas primeiras navegações pelos *campi*, no conjunto de formações de competências transversais necessárias ao seu sucesso académico e profissional. Um dos principais meios de comunicação é o correio eletrónico, o que apresenta algumas limitações no seu alcance, como o facto de muitos dos novos estudantes não verificarem a sua caixa de correio eletrónico nas primeiras semanas de aulas.

O desenvolvimento da *App* PT-UA visa contribuir com uma forma de comunicação mais próxima e adequada aos jovens estudantes. Através de uma interface baseada em rede social virtual, estes podem colocar as suas dúvidas, consultar todos os eventos que estão ou irão decorrer nos *campi*, recorrer a um *chat* em tempo real para esclarecer dúvidas ou obter informações, além do acesso a um mapa com navegação *GPS* para os guiar nos *campi*. Os utilizadores da aplicação podem comentar e dar *like* nos seus *posts* ou de outros utilizadores, estimulando o diálogo. Esta aplicação pretende que o contacto entre todos os participantes seja de forma continuada, para que estimule ligações mais fortes e dinâmicas.

O PT-UA atua em vários departamentos e escolas politécnicas da UA, sendo que cada unidade orgânica, e dentro de cada uma, cada curso tem necessidades e dificuldades específicas, que necessitam de ser tidas em conta. A *App* PT-UA pretende vir a disponibilizar uma segmentação da comunidade geral, para que cada curso e/ou unidade orgânica tenha a sua mini comunidade, de forma a dar uma resposta mais eficaz e específica. Nomeadamente, a existência de eventos ou

formações que apenas são pertinentes para aquele curso ou unidade orgânica em específico. A aplicação proposta permite fazer parte de uma comunidade, mas apenas de uma comunidade geral, sendo os seus eventos comuns a todos utilizadores.

O PT-UA disponibiliza ainda um conjunto de formações específicas para os mentores e tutores, para que estes consigam desempenhar melhor o seu papel, através de competências úteis como gestão de tempo, de *stress*, e das relações interpessoais. Na *App* PT-UA, na parte dos eventos, pretende-se que estejam presentes todos os tipos de eventos, seja as festas académicas, seja ações de formação. Outra vantagem da *App* PT-UA é a capacidade que os utilizadores têm de consultar de forma rápida e prática as informações que necessitam, como os eventos, publicações de outros estudantes, *chat*, ou até mesmo acesso ao mapa de navegação *GPS*, tudo isto através do seu *smartphone*.

4.2 Implementação

Em simultâneo ao desenvolvimento da *App* do PT-UA foi desenvolvido uma versão Web, constituída por uma plataforma onde os utilizadores podem usufruir de algumas das funcionalidades disponíveis na versão *mobile*. Este subcapítulo da Implementação está subdividido em duas partes, a primeira parte dedicada à versão *mobile*, e a segunda à versão Web.

4.2.1 Versão *App* smartphone

No desenvolvimento da *App* PT-UA foi utilizado o *react native* em conjunto com o Redux. Uma aplicação possui diversos estados, que sofrem alterações, seja quando um utilizador clica num botão, ou quando os dados mudam. Quanto mais complexa a aplicação, mais complicado se torna gerir os estados de forma eficiente e de atribuir responsabilidades. O Redux surgiu para resolver esses problemas, sendo que todos os dados são armazenados numa única *store*. No Redux, as *actions* são ativadas quando ocorre um pedido de alteração de estado, como o clique de um botão. A *action* regista o tipo (*type*) de ação (*action*) e pode também registar os valores armazenados em *input boxes* (*payload*). A função principal das *actions* é enviar o *type* e *payload* aos *reducers*. Os *reducers* são funções puras (*pure functions*), ou seja, dando o mesmo *input* à função está retornará o mesmo *output*. Quando ocorre uma *action*, esta é enviada para todos os *reducers*, sendo que só ocorre modificação quando um dos *reducers* tem previsto um tipo de ação para esse *action type*. Quando um *reducer* não tem previsto um tipo de ação para a *action*, este retorna apenas o estado. O *reducer* recebe o *payload* da *action* e cria um novo estado com as alterações. Ao fazer a alteração a *store* informa o *react* da alteração e este atualiza os dados na aplicação, e novo conteúdo é apresentado aos utilizadores. Este tipo de arquitetura permite atribuir responsabilidades, permite a

separação de preocupações, tornando-se fácil saber quem foi o responsável por alterar o estado, facilitando o *debugging*.

Ao entrar na aplicação pela primeira vez, os utilizadores são confrontados com duas possibilidades: (1) a de Entrar ou (2) a de Registar conta de utilizador. Ao clicar em Registar é-se direcionado para o ecrã de registo, onde é pedido para inserir o nome, o *e-mail*, a senha e o cargo (estudante, mentor, tutor). Em alternativa é dada a possibilidade de registo com a conta do Facebook, que irá guardar dados como nome, *e-mail*, faltando apenas ao utilizador depois preencher com o seu cargo. Na Figura 13 é representado o ecrã de *landing screen* (primeiro ecrã da aplicação) e o ecrã de registo.

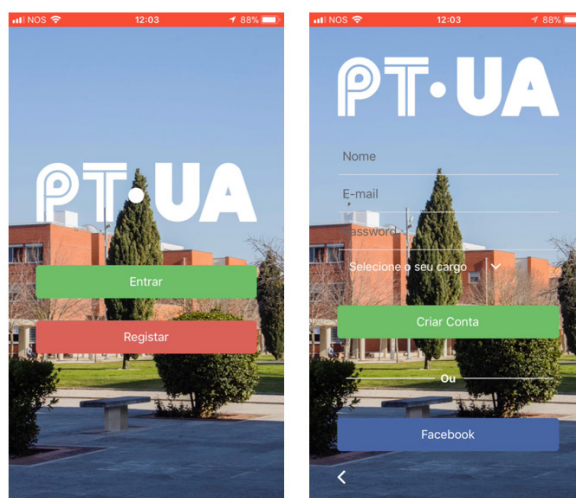


Figura 13 – *Landing screen* da App PT-UA e ecrã de registo, respetivamente.

Na escolha do cargo aparece uma caixa de seleção, como se pode observar na Figura 14. Existem três opções: estudante, mentor ou tutor. Após o registo, os utilizadores recebem um alerta a informar que o registo foi efetuado com sucesso (Figura 14).

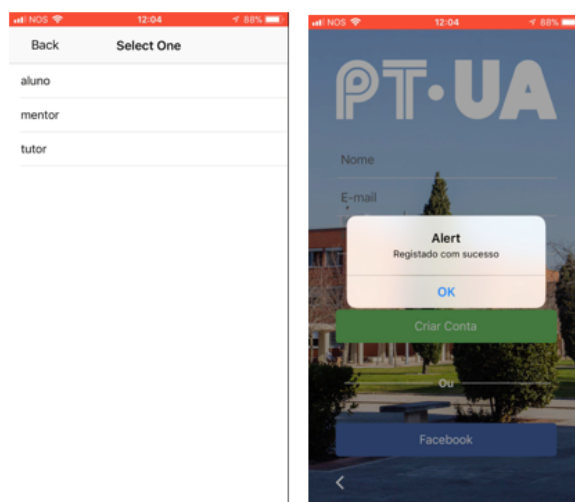


Figura 14 - Seleção do cargo de utilizador e o feedback após o registo, respetivamente.

No ecrã de Entrar, os utilizadores necessitam preencher corretamente os campos com o seu *e-mail* de utilizador e senha ou entrar com a sua conta de Facebook. Existe um sistema de validação com *feedback* visual para o utilizador, como se pode ver na Figura 15. O utilizador só consegue carregar no botão Entrar quando os campos estão devidamente preenchidos.

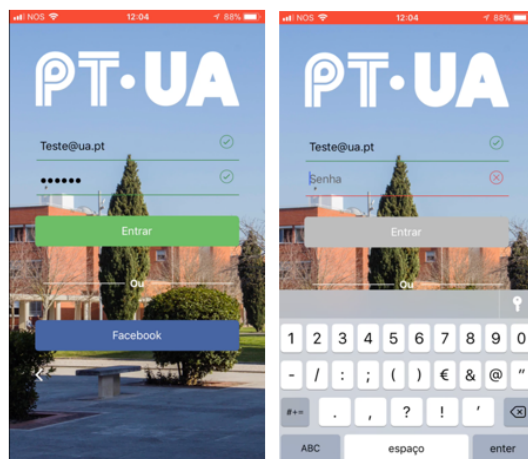


Figura 15 – *Feedback* visual durante o processo de *login* no ecrã Entrar.

Após o utilizador entrar, é confrontado com o ecrã de Comunidade (Figura 16), que, no momento dos testes de usabilidade, designava-se por Notícias, e onde estão presentes os *posts* criados por todos os utilizadores, e onde estes podem dar *like*, ver os comentários e comentar. No topo desse ecrã, o utilizador pode criar uma notícia, carregando no botão ‘criar notícia’, aparecendo um painel onde o utilizador pode inserir o título, corpo da notícia, e caso este pretenda, adicionar uma imagem.

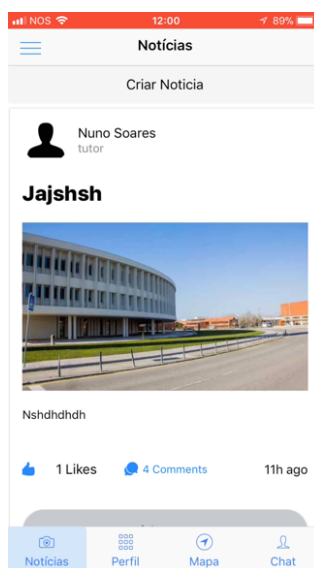


Figura 16 – Ecrã Notícias.

Na Figura 17 é apresentada uma imagem onde se pode observar o ecrã apresentado ao utilizador no momento de inserir um comentário, e onde este consegue ver o número de *likes*, e os comentários já existentes.

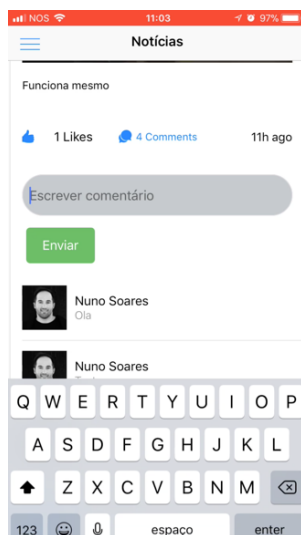


Figura 17 – Inserção de um comentário a um dos *posts* do ecrã Comunidade.

Ao clicar no menu inferior ou no menu lateral em *chat*, o utilizador é encaminhado para o ecrã de *chat*. Para implementar o *chat* foi necessário utilizar o *pusher*, e estabelecer a comunicação entre o *backend* em *laravel* e o *pusher* para que fosse possível gravar as mensagens na base de dados. Foi necessário também estabelecer a ligação entre o *react native* e o *pusher* para que fosse possível ao *react* escutar as alterações em tempo real. Para isso é necessário implementar a comunicação por *sockets*. Na parte do *backend* *Laravel* foi necessário configurar o *broadcast event* do *Laravel* para que fosse possível criar *channels*. Na Figura 18 é possível observar a inserção de uma mensagem no ecrã de *chat*.



Figura 18 – Inserção de uma mensagem de texto no ecrã de *Chat*.

Ao clicar no menu inferior ou no menu lateral em Mapa, o utilizador é encaminhado para o ecrã de Mapa. Para implementar o modo de Navegação GPS da aplicação foi necessário instalar um pacote (*package*) *react-native-maps* que facilita a incorporação do GPS e mapas. Foi necessário desenvolver os métodos específicos, e adicionar as coordenadas de cada ponto de interesse. Ao clicar no botão Localizar-me, permite obter a localização do utilizador. O utilizador pode escolher um dos Departamentos/Escolas, que ao serem seleccionados indicam a rota a seguir. A opção de Localizar-me apenas funciona após o utilizador ter dado permissão para que a aplicação consiga aceder aos seus dados referentes à localização. É possível visualizar o ecrã de Navegação e as suas funcionalidades principais, já referidas (Figura 19).

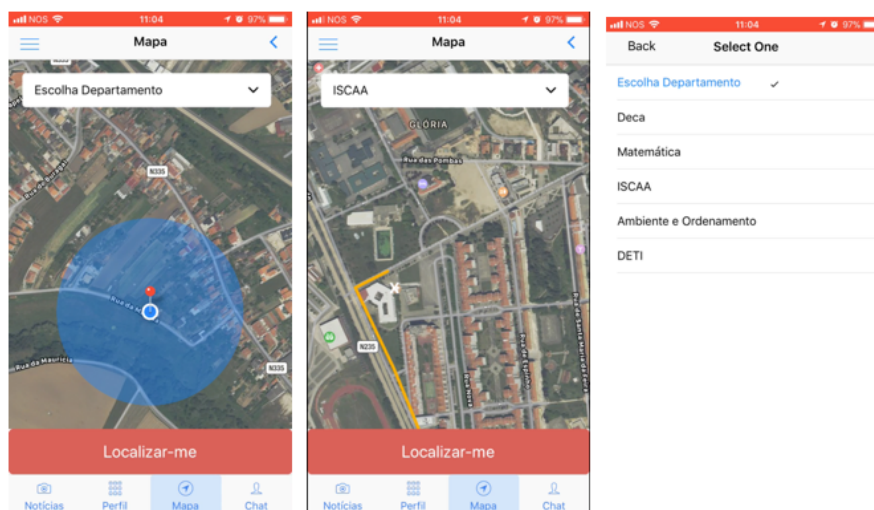


Figura 19 – *Screenshots* do ecrã Mapa e suas principais funcionalidades.

Ao clicar no menu inferior ou menu lateral em Perfil, o utilizador é encaminhado para o ecrã de perfil de utilizador e onde este pode consultar os seus dados de utilizador e modificar os seus dados. Ao mudar a imagem de perfil, o utilizador pode cortar a imagem. Podem visualizar-se vários *screenshots* do ecrã perfil e as suas funcionalidades (Figura 20).

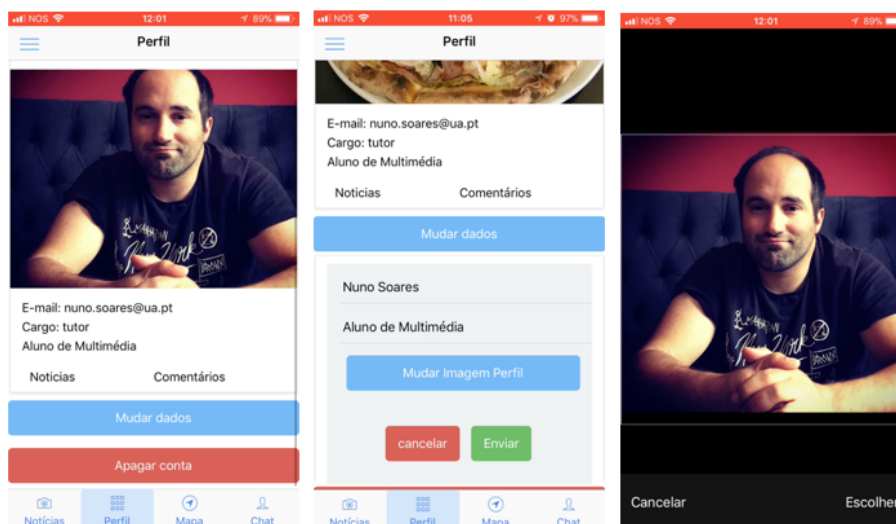


Figura 20 – *Screenshots* do ecrã de Perfil e a alteração dos dados do utilizador.

Ao carregar no menu lateral e clicar em Eventos, o utilizador é direcionado para o ecrã de Eventos onde este pode consultar os eventos disponíveis atualmente, e adicionar novos eventos, após o preenchimento dos dados, como o título do evento, conteúdo do evento, local, data e uma imagem, caso o utilizador pretenda adicionar uma imagem do evento. Na Figura 21 pode observar-se o ecrã de Eventos durante a listagem e a criação de eventos.

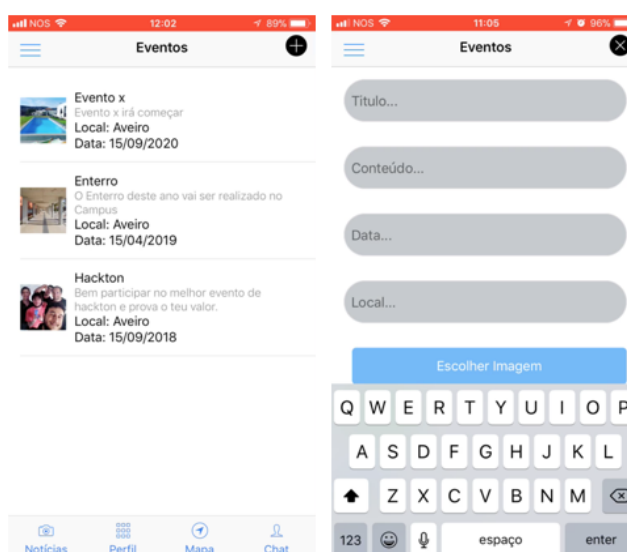


Figura 21 – Ecrã de Eventos com a listagem e a criação de um evento, respetivamente.

Como não foi feita a compra das licenças de desenvolvedor da *Apple (iOS)* e da *Google (android)*, de forma a poder ser testável o código nos telemóveis, foi necessário recorrer ao *EXPO XDE*. O *EXPO XDE* permite também escrever código uma única vez e converte posteriormente o código para *iOS* e para *android*.

Na parte gráfica da aplicação foi feito recurso ao *Native Base*, que disponibiliza um conjunto de componentes de *UI* compatíveis com *android* e *iOS*. O *Native Base* foi utilizado para os botões, *input boxes*, menu inferior e lateral, assim como todos os ícones.

O desenvolvimento desta aplicação foi complexo, sendo que, por exemplo, a funcionalidade dar um *like* gera um pedido (*query*) à base de dados que obriga a um cruzamento de dados de 3 tabelas diferentes.

Relativamente ao processo de *login* através da aplicação móvel, é feito através de um pedido (*request*) *http* feito pelo *react native* à *API* do *Laravel*. O *Laravel* recebe esse pedido do tipo *POST* e compara os dados com os dados na base de dados, retornando uma resposta ao *react native*. A resposta pode ser “dados não coincidem”, não ocorrendo a autenticação do utilizador, ou pode ser “*login* feito com sucesso”, dando acesso aos restantes ecrãs da aplicação. Sem autenticação feita, os utilizadores apenas podem aceder ao ecrã de entrada, ao de *login* e de registo de utilizador.

No processo de registo de utilizador, é feito um pedido *http* por parte do *React Native* à *API* do *Laravel*. O *Laravel* recebe esse pedido do tipo *POST*, e insere os dados na base de dados, retornando uma resposta ao *React Native*. A resposta pode ser “utilizador criado com sucesso” ou retorna uma mensagem de erro.

No ecrã de Perfil é possível ver os dados de utilizador. Ao clicar no botão de editar perfil é possível editar os dados de perfil e imagem de utilizador, e ao clicar no botão de apagar conta, é possível apagar a conta de utilizador. No editar perfil é feito um pedido *HTTP* à *API* do *Laravel*, processando este a alteração dos dados junto da base de dados, e ou o *upload* da imagem de perfil. Ao alterar com sucesso, o ecrã atualiza automaticamente, visualizando as alterações efetuadas. No editar perfil não é obrigatório alterar todos os dados, basta apenas modificar os que se pretende alterar.

No ecrã da Comunidade, para criar uma publicação basta carregar no botão criar publicação, tendo de introduzir o título, descrição e caso pretenda, uma imagem. Ao clicar em enviar, é feito um pedido *http* do *react native* à *API* do *Laravel*, tratando este do *upload* da imagem caso haja, e da inserção dos restantes dados na base de dados.

No ecrã dos Eventos, é possível visualizar os eventos, e carregando no botão “+” no canto superior direito, é possível adicionar novos eventos, tendo o utilizador de preencher com os dados como local, data de realização do evento, nome, descrição e caso pretenda adicionar uma imagem. Ao

carregar enviar, é feito um pedido *http* do React Native à API do Laravel, tratando este do *upload* da imagem caso haja, e da inserção dos restantes dados na base de dados.

Para aceder à biblioteca de imagens para *upload* de imagens ou à localização do utilizador, é necessário um conjunto de autorizações, sendo estas alterações tratadas pelo EXPO, sendo que após o utilizador aceitar, este não é mais confrontado com esse pedido de autorização.

Apresenta-se na Figura 22 um esquema de toda a comunicação entre a App em React Native e Redux e a API desenvolvida em Laravel.

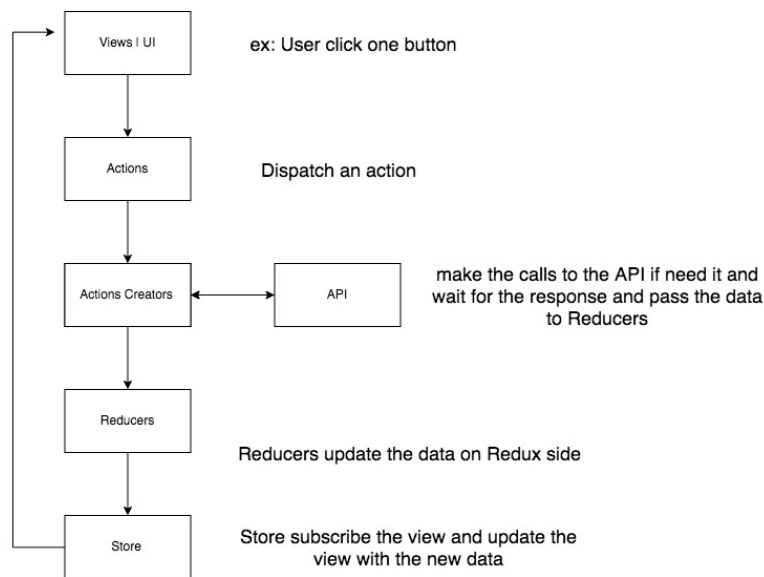


Figura 22 – Esquema do funcionamento do *Redux* na App PT-UA.

A Figura 23 apresenta um esquema que sintetiza as formas e percurso de utilização da aplicação.

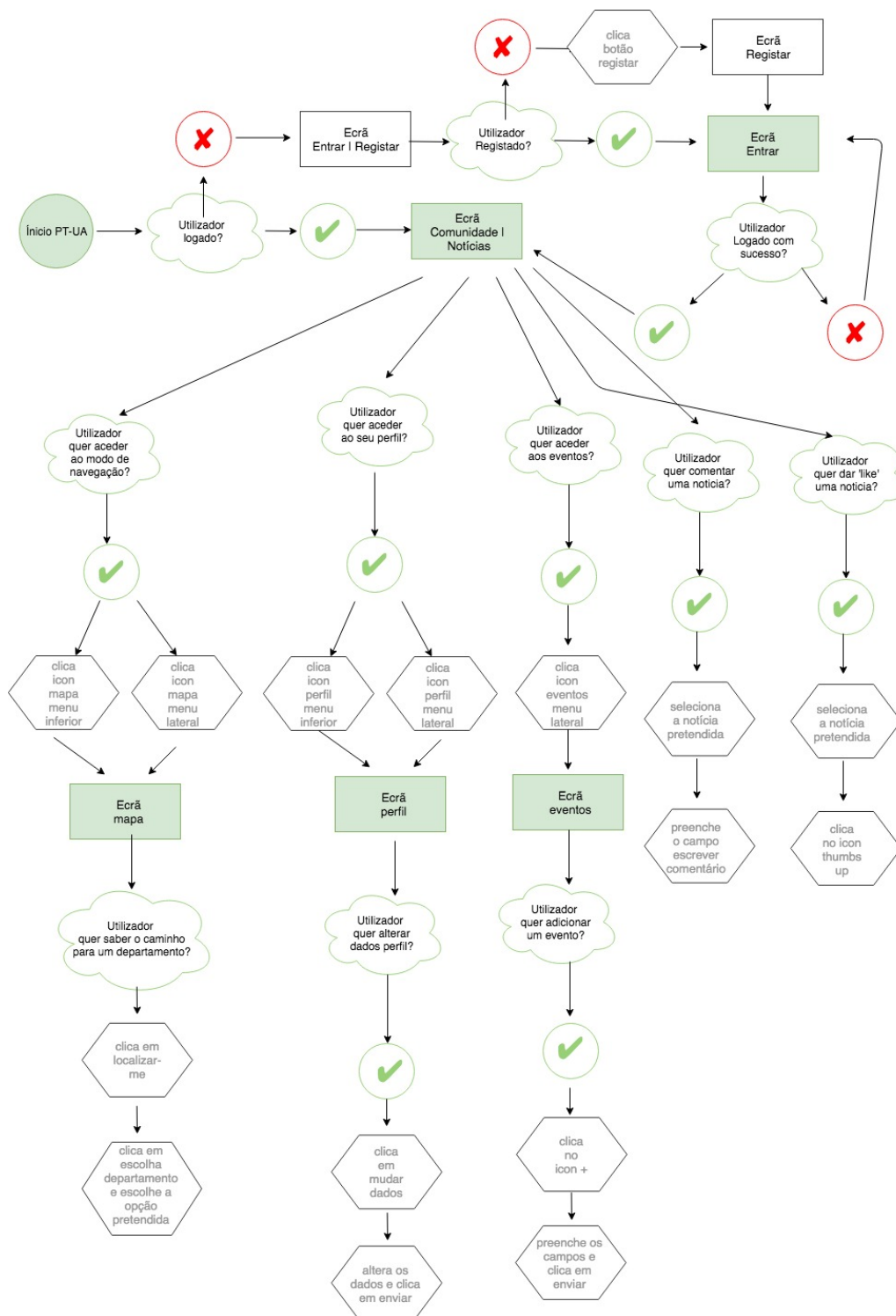


Figura 23 – Esquema elucidativo navegação do utilizador na App PT-UA.

Este tipo de esquema é muito útil para encontrar potenciais problemas na experiência de utilização, mesmo antes do desenvolvimento da mesma, ou após o desenvolvimento da mesma, para se procurar formas de simplificar o percurso do utilizador, eliminando passos desnecessários e que podem ser potencialmente aborrecidos para os utilizadores. Se ocorrer um crescimento da aplicação

e passarem a existir vários elementos com diferentes responsabilidades como design, código, gestão, etc, fica mais fácil a colaboração, uma vez que através deste esquema todos conseguem compreender qual o percurso do utilizador na aplicação.

4.2.2 Versão plataforma *web*

A parte da plataforma *web* permite aos utilizadores usufruir de muitas das funcionalidades oferecidas na aplicação móvel (App). Foi desenvolvida totalmente em Laravel, obrigando à criação de todo o aspeto gráfico da mesma. Não foi utilizado nenhuma *framework css*, sendo este todo desenvolvido de raiz. O seu aspeto gráfico foi feito de forma a ser muito idêntico ao da App para que os utilizadores não fiquem baralhados. A plataforma não oferece o modo de navegação GPS nem a possibilidade de comentar e dar gostos nas publicações. É possível visualizar o ecrã Comunidade (Figura 24), que na altura da realização dos testes se chamava de Notícias.

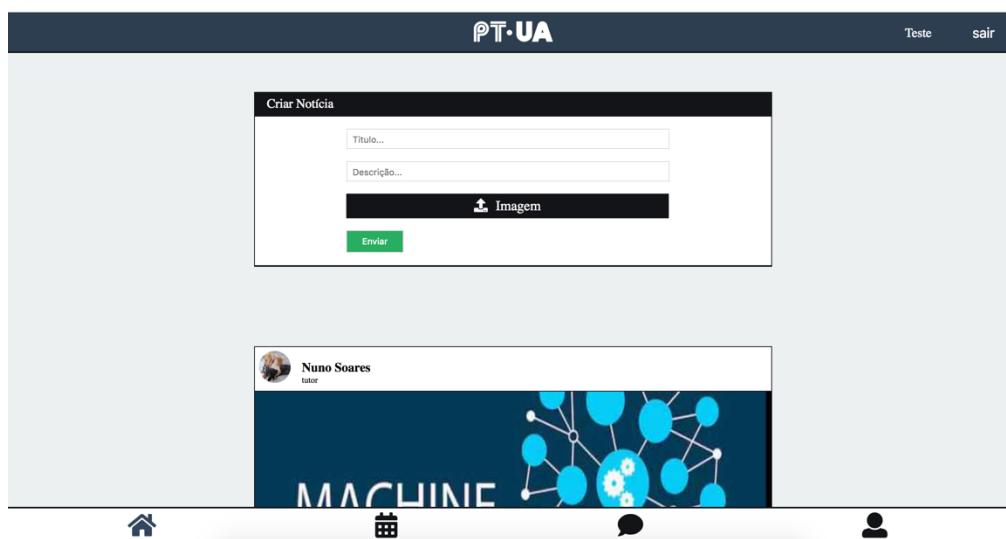


Figura 24 – Ecrã Notícias da versão Web, destinada ao uso em Desktop.

Na figura 25 é possível visualizar a versão *web* do *Chat*, e algumas das conversações realizadas.

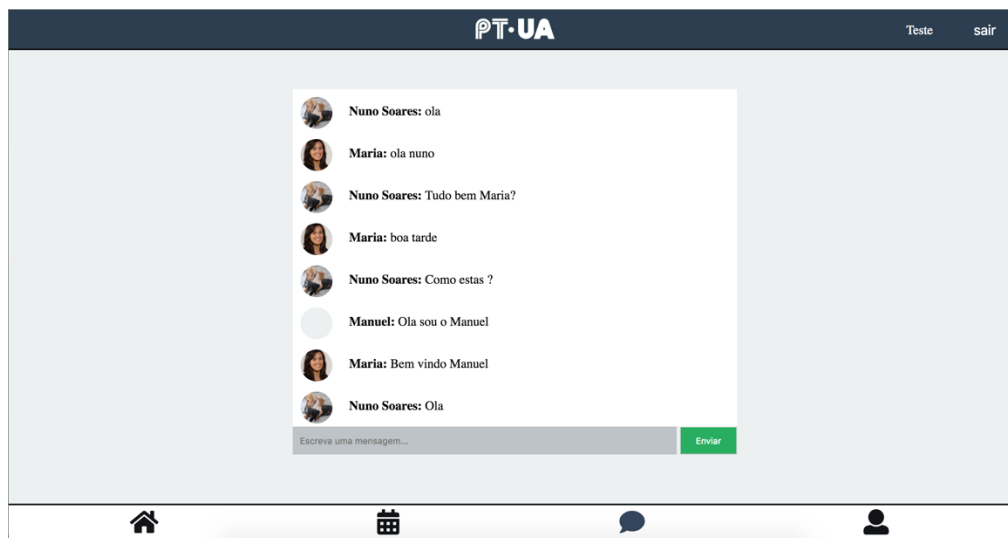


Figura 25 – Ecrã de *Chat* da versão *Web*.

Na Figura 26 é possível visualizar o ecrã de Eventos e o seu formulário de criação.

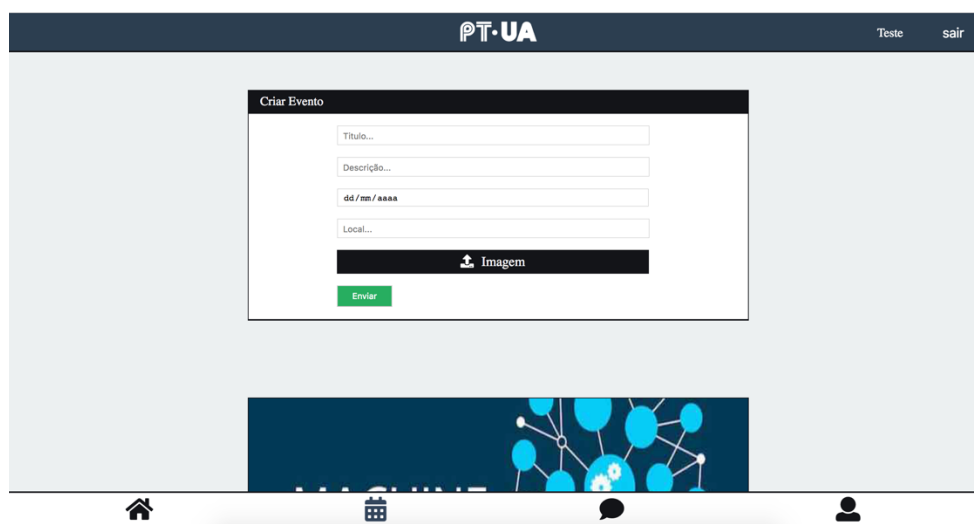


Figura 26 – Ecrã de Eventos com o formulário de criação de eventos.

Pode visualizar-se o ecrã de Perfil (Figura 27) da versão Web, e visualizar o modal que surge durante a alteração dos dados de perfil (Figura 28).

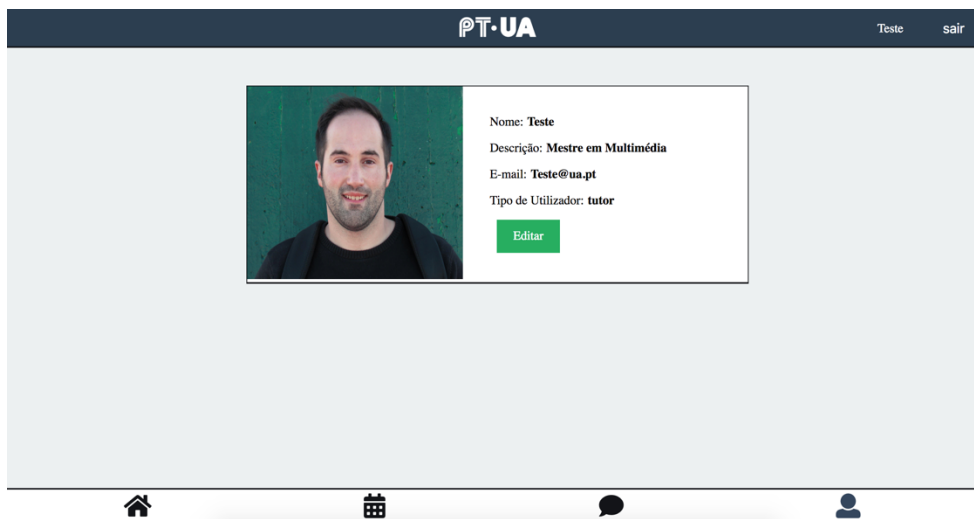


Figura 27 – Ecrã de Perfil da versão Web.

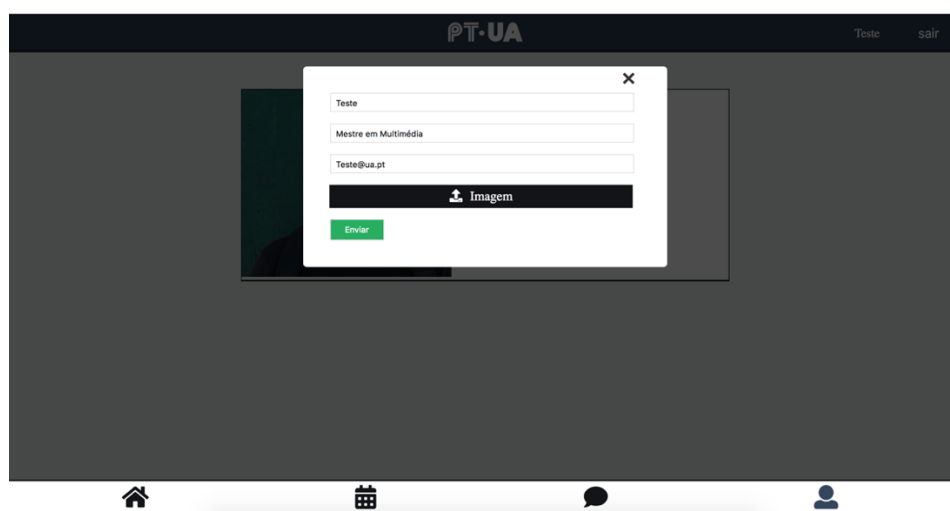


Figura 28 – Alteração dos dados de utilizador no ecrã de Perfil, versão Web.

No modo de Chat é possível uma conversação pública com todos os utilizadores da App e da plataforma Web. É uma ótima forma de mentores e estudantes interagirem. Quando alguém envia uma mensagem de texto no *Chat*, é emitido um evento ao *Laravel Broadcast Event* e este envia a mensagem para o *pusher*, informando assim todos os utilizadores que estejam conectados no *Chat*, atualizando automaticamente com as novas mensagens, em tempo real.

4.3 Testes de usabilidade

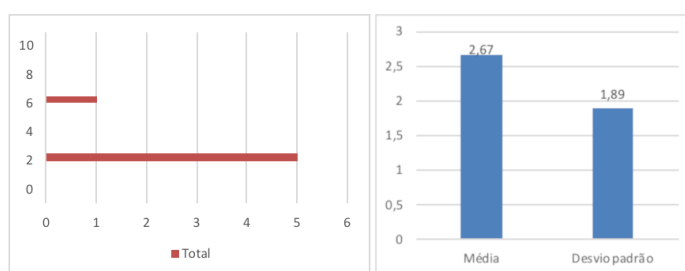
Após o desenvolvimento e implementação do protótipo foram realizados testes de usabilidade, a potenciais utilizadores, conforme condições previamente identificadas no Capítulo 3.

Os testes incluíram um conjunto de tarefas, e, após a sua conclusão, os participantes foram confrontados com um conjunto de questões acerca de como se sentiam na utilização dos vários ecrãs e o sentimento geral na utilização da aplicação e no grau de dificuldade na navegação entre os vários ecrãs da App. Para as questões relacionadas com o sentimento da utilização da App utilizou-se o método *Emocards*, com uma escala de zero a dez, onde zero correspondia a muito feliz e dez a muito infeliz.

As tarefas consistiam em criar uma nova conta de utilizador, fazer *login*, *logout*, criar um novo *post*, comentar um *post*, utilizar o mapa para navegar para um departamento à sua escolha, alterar a suas informações de perfil, etc.

Relativamente ao grau de satisfação geral na utilização da App, 5 dos participantes consideraram-se muito satisfeitos, e 1 participante considerou a sua experiência como nem satisfatória nem insatisfatória. A média de satisfação foi de 3, e o desvio padrão de 2 valores como se pode ver na Figura 29.

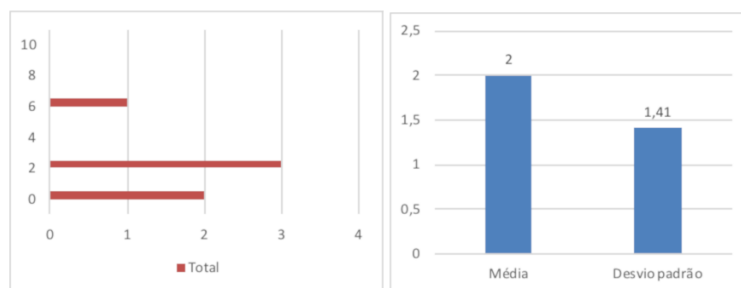
Figura 29 – Grau de satisfação geral na utilização da App.



Legenda: À esquerda o eixo dos y corresponde aos valores da escala dos *Emocards*, e o eixo dos x à frequência absoluta. À direita é apresentado a média e desvio padrão.

Sobre o grau de satisfação relativamente à navegação inferior e lateral, 2 participantes indicaram sentirem-se bastante satisfeitos (deram valor zero), 3 participantes muito satisfeitos (deram valor 2), e 1 dos participantes não se sentiu nem satisfeito nem insatisfeito. A média de satisfação obtida foi de 2 valores, o que corresponde a um resultado muito satisfatório. O desvio padrão foi de 1 valor. Estes resultados podem ser observados na Figura 30.

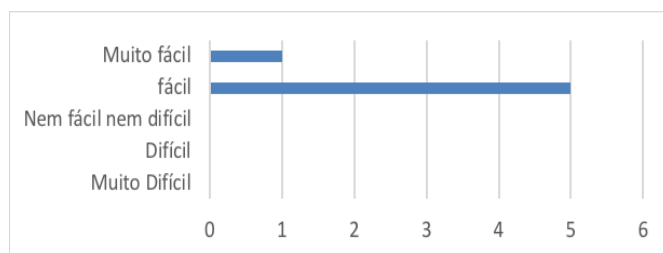
Figura 30 – Grau de satisfação referente à navegação inferior e lateral



Legenda: À esquerda o eixo dos y corresponde aos valores da escala dos *Emocards*, e o eixo dos x à frequência absoluta. À direita é apresentado a média e desvio padrão.

Na avaliação do grau de dificuldade sentido pelos participantes relativamente à navegação do menu lateral e inferior, 5 consideraram fácil a navegação e 1 dos participantes considerou muito fácil, como pode ser observado na Figura 31.

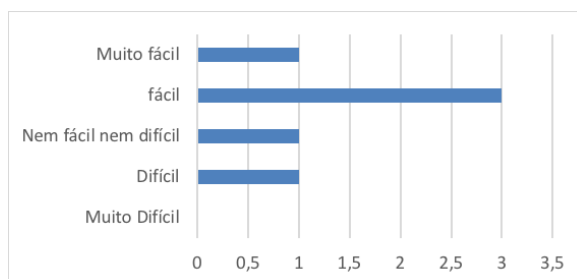
Figura 31 – Grau de dificuldade relativamente à navegação (inferior e lateral).



Legenda: O eixo de x corresponde às frequências absolutas.

Sobre o grau de dificuldade relativamente à criação de uma nova conta de utilizador, 3 participantes consideram fácil a criação de uma nova conta de utilizador, e nos restantes participantes, 1 considerou muito fácil, outro considerou nem fácil nem difícil, e 1 considerou difícil, como pode ser observado na Figura 32.

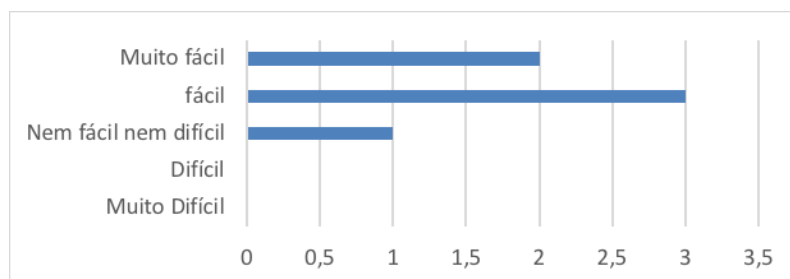
Figura 32 – Grau de dificuldade relativamente à criação de uma nova conta.



Legenda: O eixo de x corresponde às frequências absolutas.

Sobre o grau de dificuldade relativamente à criação de uma nova notícia, 3 participantes consideraram fácil, 2 consideraram muito fácil, e 1 participante considerou nem fácil nem difícil, como pode ser observado na Figura 33.

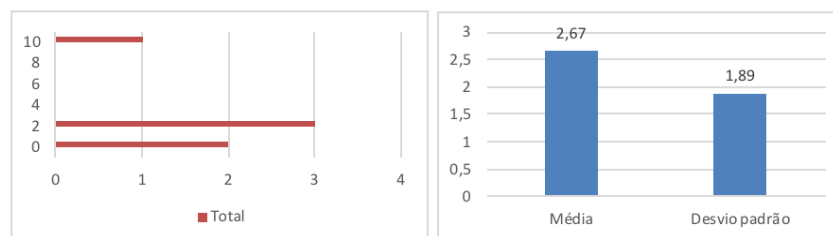
Figura 33 – Grau de dificuldade relativamente à criação de uma nova notícia.



Legenda: O eixo de x corresponde às frequências absolutas.

Relativamente ao grau de satisfação do *feed* de notícias, e na possibilidade de adicionar novos conteúdos, comentar e dar gosto, 2 participantes consideram-se bastante satisfeitos, 3 consideram-se muito satisfeitos, e 1 participante considerou-se muito insatisfeito. A média foi de 3 valores, o que é um resultado francamente positivo. Estes resultados podem ser observados na Figura 34.

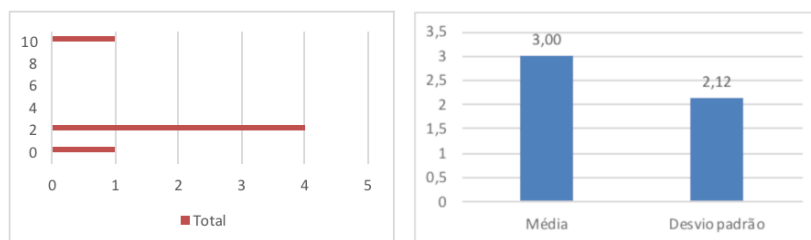
Figura 34 – Grau de satisfação do feed de notícias e suas funcionalidades.



Legenda: À esquerda o eixo dos y corresponde aos valores da escala dos *Emocards*, e o eixo dos x à frequência absoluta. À direita é apresentado a média e desvio padrão.

Sobre o grau de satisfação relativamente ao ecrã de perfil e na possibilidade de alterar os dados de utilizador, 4 participantes consideraram-se muito satisfeitos (deram valor 2), 1 participante mostrou-se bastante satisfeito (deu valor 0), e 1 participante mostrou-se muito insatisfeito. Estes resultados podem ser observados na Figura 35.

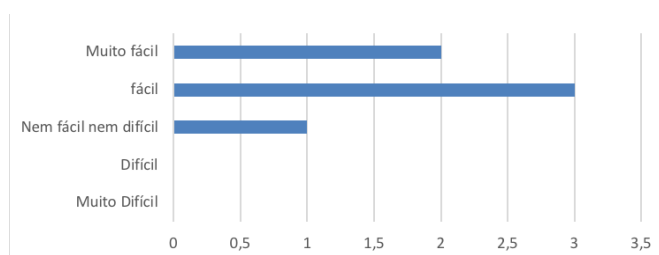
Figura 35 – Grau de satisfação relativamente ao ecrã de perfil e edição de dados.



Legenda: À esquerda o eixo dos y corresponde aos valores da escala dos *Emocards*, e o eixo dos x à frequência absoluta. À direita é apresentado a média e desvio padrão.

Relativamente ao grau de dificuldade na edição dos dados de perfil, 3 participantes consideraram fácil, 2 participantes consideraram muito fácil, e 1 participante considerou não ser fácil nem difícil, como pode ser observado na Figura 36.

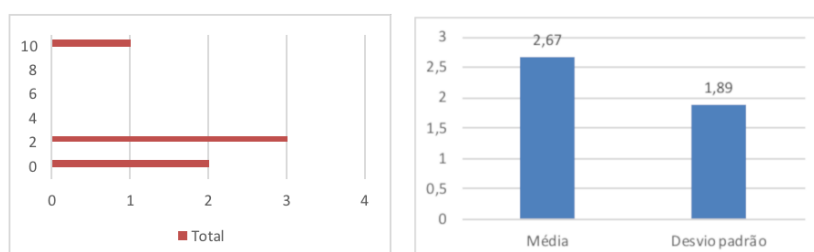
Figura 36 – Grau de dificuldade na edição dos dados de perfil.



Legenda: O eixo de x corresponde às frequências absolutas.

Sobre o grau de satisfação relativamente ao ecrã de *chat*, 3 participantes consideram-se muito satisfeitos, 2 praticantes bastante satisfeitos, e 1 participante sentiu-se muito insatisfeito. A média de satisfação desta questão foi de 3 valores, e o desvio padrão foi de 2 valores. Estes resultados podem ser observados na Figura 37.

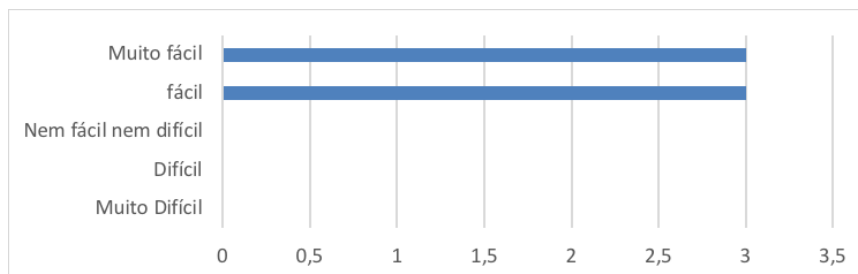
Figura 37 – Grau de satisfação relativamente ao ecrã de *chat*.



Legenda: À esquerda o eixo dos y corresponde aos valores da escala dos *Emocards*, e o eixo dos x à frequência absoluta. À direita é apresentado a média e desvio padrão.

Quanto ao grau de dificuldade relativamente à utilização do *Chat*, 3 participantes consideraram muito fácil, e 3 participantes consideraram fácil, como pode ser observado na Figura 38.

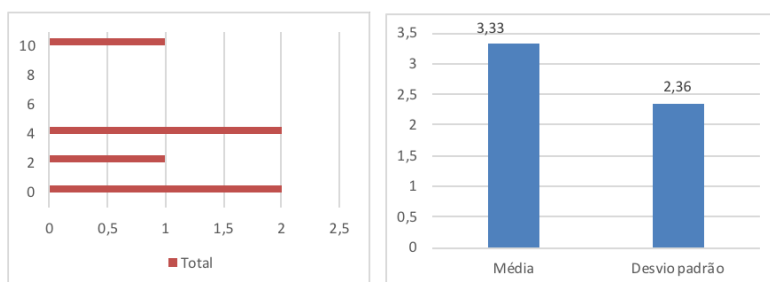
Figura 38 – Grau de dificuldade da utilização do *Chat*.



Legenda: O eixo de x corresponde às frequências absolutas.

Relativamente ao grau de satisfação do ecrã de mapa | navegação, 2 participantes deram cotação zero, 2 participantes deram cotação quatro, 1 participante deu cotação dois, e 1 participante deu cotação dez. A média obtida foi de 3 valores, e o desvio padrão foi de 2 valores. Estes resultados podem ser observados na Figura 39.

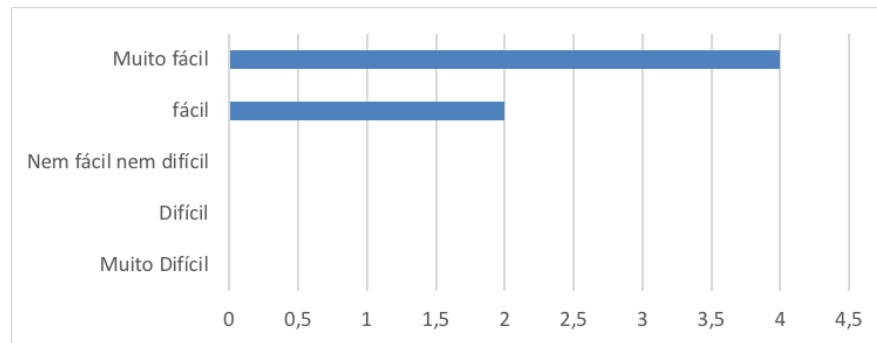
Figura 39 – Grau de satisfação do ecrã de Mapa | Navegação.



Legenda: À esquerda o eixo dos y corresponde aos valores da escala dos *Emocards*, e o eixo dos x à frequência absoluta. À direita é apresentado a média e desvio padrão.

Sobre o grau de dificuldade relativamente à utilização do mapa de navegação, 4 participantes consideraram muito fácil a navegação, e 2 participantes consideraram fácil, como pode ser observado na Figura 40.

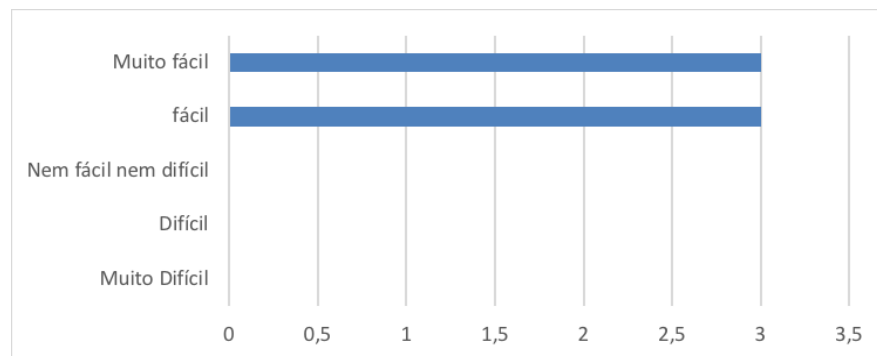
Figura 40 – Grau de dificuldade relativamente do mapa de Navegação.



Legenda: O eixo de x corresponde às frequências absolutas.

Relativamente ao grau de dificuldade na criação de um novo evento, 3 pessoas consideraram muito fácil, e os restantes 3 participantes consideraram fácil, como pode ser observado na Figura 41.

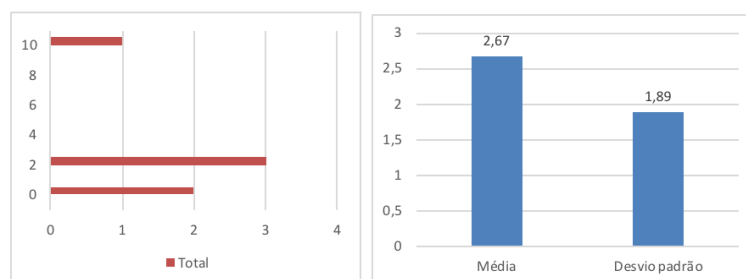
Figura 41 – Grau de dificuldade relativamente à criação de um novo evento.



Legenda: O eixo de x corresponde às frequências absolutas.

Sobre o grau de satisfação relativamente ao ecrã de eventos e na possibilidade de adicionar novos conteúdos, 3 participantes consideraram-se muito satisfeitos (deram valor dois), 2 participantes consideraram-se bastante satisfeitos (deram valor zero), e 1 participante sentiu-se muito insatisfeito. A média foi de 3 valores, e o desvio padrão de 2 valores (Figura 42).

Figura 42 – Grau de satisfação relativamente ao ecrã de eventos e criação de novos conteúdos.



Legenda: À esquerda o eixo dos y corresponde aos valores da escala dos *Emocards*, e o eixo dos x à frequência absoluta. À direita é apresentado a média e desvio padrão.

Sobre as funcionalidades que os participantes achavam importante adicionar (questão aberta), foram sugeridas:

- existência de um calendário no campo de inserção da data no ecrã de criação de eventos;
- existir feedback visual quando os elementos estiverem a carregar, nomeadamente ao fazerem pedidos de dados à *API*;
- a importância de ser possível retirar gostos e apagar comentários;
- no ecrã de *chat* ser possível para criar grupos de conversa e fazer conversas privadas entre utilizadores;
- a criação de eventos só deveria estar disponível para mentores e tutores;
- existir uma *mini cloud* onde desse para partilhar ficheiros;
- no ecrã de perfil existir mais informações como a disponibilidade dos mentores e tutores;
- dar gosto com duplo toque, semelhante ao que é disponibilizado pelo *Instagram*;

Sobre os aspetos da aplicação que os participantes menos gostaram, e se estes mudariam algo (questão aberta), as alterações sugeridas foram:

- o *chat* deveria ser de cima para baixo;
- os ícones poderiam ter um texto a acompanhar para facilitar a sua identificação;
- no ecrã de *chat*, a caixa de texto responsável por adicionar novas mensagens deveria ser fixa e o ícone de enviar deveria apenas conter um “+”;
- o menu inferior deveria conter apenas o mapa, perfil, *chat* privado e *logout*;

A salientar que 1 dos participantes, aparentemente, confundiu a escala do método de *Emocards*, uma vez que este colocou em muitas das questões nível de satisfação dez (muito insatisfeito) contudo nas questões relativas ao grau de dificuldade, este considerou muito fácil. Não referiu nada relativamente ao que o fez sentir-se muito insatisfeito.

Também de referir que da observação do investigador ficou a percepção que no geral os participantes conseguiam realizar as tarefas de uma forma rápida, sem ficarem confusos, e fizeram referência às similaridades entre a *App* e o *Instagram*, nomeadamente no ecrã de Comunidade.

A seguir apresentam-se as sugestões de melhoria a implementar, resultantes dos resultados dos testes de usabilidade apresentados anteriormente e outras, entretanto identificadas.

4.4 Melhorias implementadas e a implementar

Após e durante a realização dos testes de usabilidade surgiram algumas sugestões de melhoria interessantes de implementar. A seguir irá ser apresentado um resumo com as sugestões de melhoria. De todas as sugestões foram implementadas algumas, referidas no final deste subcapítulo. Não foram implementadas todas as sugestões, uma vez que o processo de desenvolvimento é algo demorado e complexo, não existindo tempo para concluir tudo.

Atualmente não existe nenhum servidor de *e-mail* implementado, necessário para que os utilizadores possam confirmar a sua conta, embora toda essa lógica esteja já implementada, mas encontrando-se comentada, para que seja possível criar conta e testar prontamente. Durante a realização dos testes, todos os utilizadores tinham os mesmos privilégios, tendo sido referido a necessidade de restringir algumas opções como a criação de eventos, para apenas os tutores conseguirem adicionar eventos. Já foram criados os métodos para que os mentores e tutores possam aceitar e validar os registos de utilizadores para os cargos de mentor e tutor, estando comentados, por ser necessário desenvolver as camadas gráficas das mesmas. Durante a realização dos testes os tutores e mentores não podiam apagar conteúdos inapropriados de estudantes e moderar as publicações e chat. Falta implementar o *chat* privado, e seria interessante adicionar a partilha de ficheiros, e vídeo chamada. O ecrã de *chat* deveria conter os utilizadores que estavam *online* no momento, assim como os *offline* e/ou ausentes. Os utilizadores participantes dos testes de usabilidade apontaram a necessidade de um motor de pesquisa ou uma forma de filtrar as postagens, para que seja possível encontrar as informações mais rapidamente. No ecrã de criação de eventos da *App* poderia surgir um calendário para os utilizadores selecionarem as datas dos eventos, e ao clicarem no campo local, surgir um mapa onde estes selecionavam o local, ficando posteriormente disponível para os outros utilizadores puderem usar a navegação até ao local. Seria interessante implementar um sistema de notificações para notificar os utilizadores quando outro utilizador enviasse uma mensagem privada, ou quando um novo evento fosse adicionado. Seria também interessante adicionar a possibilidade de identificar outros utilizadores nos comentários, e a possibilidade de os utilizadores partilharem as postagens nos seus perfis de *facebook*. Durante as realizações dos testes só era possível aceder ao perfil do utilizador logado, tendo sido referido que seria interessante dar para ver os perfis de outros utilizadores, e os perfis dos tutores ter às suas

áreas de especialização e a sua disponibilidade. Atualmente falta implementar os campos de especialização e disponibilidade nos ecrãs de perfil dos tutores. Foi sugerida a possibilidade de os tutores adicionarem pontos de interesse no ecrã de mapa.

A Tabela 8 resume as sugestões de melhoria.

Resumo das sugestões de melhoria
Implementar servidor de e-mail para ser possível envio de e-mail para confirmação de conta.
Desenvolver a lógica na aplicação para que cada tipo de utilizador só detenha determinados privilégios.
Possibilidade dos tutores e mentores validarem pela aplicação as novas contas com tipo de utilizador tutor ou mentor.
Possibilidade dos tutores e mentores puderem apagar conteúdos e moderarem as publicações.
Implementar chat privado, partilha de ficheiros e vídeo chamada.
Ecrã de chat deveria indicar qual o estado do utilizador (online, ausente, offline).
Implementar motor de busca para que seja possível filtrar e encontrar as informações mais rapidamente.
No ecrã de criação de eventos deveria existir um calendário para selecionar as datas dos eventos.
No ecrã de criação de eventos deveria existir um mapa para selecionar o local.
Implementar um sistema de notificações para informar os utilizadores da adição de novos eventos, ou mensagens privadas.
Possibilidade dos utilizadores partilharem os posts nos seus perfis de facebook.
Possibilidade de identificar utilizadores nos comentários.
Aceder ao perfil de outros utilizadores, sendo que o perfis dos tutores devem ter as suas áreas de especialização e sua disponibilidade.
Implementar o duplo toque em um post para dar like.
Possibilidade dos tutores adicionarem novos pontos de interesse no ecrã de mapa.
Criar ecrã de detalhe de evento.

Tabela 8 – Resumo das sugestões de melhoria.

Face a estas sugestões de melhoria procedeu-se à implementação de algumas destas, assim como a correção de alguns *erros*. O ecrã de Comunidade apresenta corretamente as imagens dos utilizadores que comentam ou criam as publicações. Ao clicar na imagem do utilizador é possível aceder ao seu perfil.

Foram corrigidos alguns erros no *login* e registo com o *Facebook*, estando agora devidamente funcional. Ao clicar em criar conta com o *Facebook*, abre uma janela do próprio *Facebook*, pedindo autorização ao utilizador para partilhar os seus dados de perfil e *e-mail* com a *App*. Ao aceitar, é devolvido à *App* as informações, aparecendo alguns campos já preenchidos, faltando apenas ao utilizador adicionar o seu cargo. Ao carregar em registar os dados de utilizador são registados na base de dados. No ecrã de Entrar, ao carregar com *login* pelo *Facebook*, abre uma janela do *Facebook* pedindo autorização ao utilizador, que ao aceitar é redirecionado para o interior da *App*, para o ecrã de Comunidade.

Foi adicionado o ecrã de detalhe de evento, ficando assim mais perceptível a imagem do evento e restantes informações. Nesse ecrã foi adicionado um botão no canto superior direito que permite voltar ao ecrã de eventos.

Foram adicionados *feedbacks* visuais, aparecendo um ícon de *loading*, enquanto é feito o carregamento dos conteúdos, provenientes da *Api*. No ecrã de *chat* foi adicionado um botão vermelho que permite ir para o fundo do *chat*, permitindo visualizar as mensagens mais recentes.

No próximo capítulo são apresentadas as conclusões finais e algumas limitações da presente dissertação.

5. Conclusões finais

Os jovens utilizam cada vez mais as redes sociais e a *Internet*, levando a uma pressão por parte das universidades para acompanharem esta mudança de comportamentos e conseguirem estar mais próximas dos seus estudantes. Com base nesta premissa, na realização desta dissertação foi possível analisar várias formas de implementar uma aplicação de suporte ao Programa de Tutoria da Universidade de Aveiro (PT-UA), nomeadamente, nas várias teorias associadas à vertente tecnológica, do *design* e do *UX*. Foi feita uma análise cuidadosa a software já existente no mercado, cujo objetivos eram similares ao desta dissertação. Na conceção dos *layouts* foram tidas em conta as principais recomendações da *Apple*, *Google*, diversos autores de *UX*, assim como boas práticas de desenvolvimento de *software*. Foram também analisadas algumas aplicações e funcionalidades, como a navegação com recurso a *GPS* para guiar os novos estudantes dentro dos *campi*, a possibilidade de existirem pequenas comunidades, troca de ficheiros e conversações online entre os estudantes e mentores e tutores.

O número de participantes (6) nos testes de usabilidade foi adequado para este tipo de avaliação e permitiu encontrar vários pontos a serem melhorados. Um conjunto de funcionalidades novas foram ainda sugeridas e poderão ser implementadas no futuro, para permitir uma melhor experiência de utilização. Não é possível generalizar os resultados obtidos para toda a população do PT-UA, pelo que, após a implementação das sugestões de melhoria, justificar-se-ia a realização de outro tipo de teste junto de uma amostra maior. A amostra deverá ser constituída por elementos de vários departamentos, e com cargos diferentes, para que os resultados não sejam enviesados, e para que estes se aproximem o mais possível da realidade.

Optou-se pela utilização de tecnologias atuais, desenvolvidas e mantidas por grandes empresas e/ou comunidades online, e que já contassem com inúmeros produtos e serviços no mercado. Por exemplo, o *react native* e *redux* são utilizados em várias aplicações, como a do *Facebook*, *Instagram*, *Airbnb*, *Udacity*. O *redux* permite uma gestão previsível dos estados dentro de uma aplicação, além de facilitar a escalabilidade à medida que novas funcionalidades possam ser adicionadas. Relativamente ao *backend*, apesar de o *laravel* ser muito utilizado, ter uma comunidade muito forte por de trás e oferecer muitas funcionalidades úteis, poderiam ter sido utilizadas outras alternativas sem que fosse necessário desenvolver tanto código. Alternativas essas como o *firebase* da *Google*, que permite o *login* com redes sociais, salvar dados e ficheiros, além de oferecer comunicações em tempo real, como quando um utilizador submete algo, todos os outros utilizadores têm a aplicação ou página atualizada com os novos dados, sem terem de fazer *refresh* na página ou aplicação. O *firebase* é utilizado atualmente em vários *websites* e aplicações devido a reduzir a complexidade de implementação de um *backend* robusto e de raiz.

Foram cumpridos os objetivos iniciais de identificar formas de desenvolver uma aplicação de suporte ao PT-UA, assim como adicionalmente se desenvolveu uma versão *web*, para que os utilizadores possam usufruir de algumas das funcionalidades fora dos seus *smartphones*.

A *App* do PT-UA permite melhorar o processo de comunicação entre todos os participantes do programa, através da comodidade de aceder às informações a partir do seu *smartphone*, a possibilidade de participar em conversações em tempo real. Facilita a navegação dos novos estudantes dentro do campus, através da disponibilização de uma funcionalidade de navegação *GPS* dentro da *App*. Oferece comodidade aos seus utilizadores com a sua facilidade em obter as informações sobre os principais eventos a decorrer na Universidade de Aveiro. Permite a colocação de publicações dos seus membros, estimulando à criação de uma comunidade online, melhorando os laços relacionais. Permite auxiliar na gestão do PT-UA, uma vez, que facilita o processo de comunicação entre todos os envolvidos. Esta *App* poderá ser útil para outras universidades com programas similares, sendo uma forma de estas estarem mais próximas dos estudantes, comunicando nos meios que estes mais utilizam, além de toda a comodidade de aceder a informação em qualquer lugar, desde que se tenha *smartphone* e acesso à *Internet*.

Em trabalhos futuros outras funcionalidades sugeridas durante a realização dos testes de usabilidade podem ser implementadas, nomeadamente, a opção de *chat* privado, a partilha de ficheiros e videochamada entre estudantes, tutores e mentores. Também a opção de motor de busca que permita filtrar os *posts* e facilitar o acesso mais rápido à informação, uma vez que se ocorrerem muitos *posts* torna-se mais difícil encontrar os dados. Outra funcionalidade que pode ainda ser adicionada é a indicação do estado de cada utilizador no *chat*, para os participantes saberem quem está *online*, ausente e *offline*. Esta funcionalidade é bastante útil, uma vez que um utilizador ao enviar uma mensagem para um utilizador que está ausente, não ficará à espera de uma resposta rápida. Pode também ser incluída a opção que permita identificar outros utilizadores em um determinado *post*. Um sistema de notificações para quando o utilizador receber uma mensagem privada, ou for identificado em determinada publicação, poderá também ser outro aspeto a incluir. No ecrã de eventos, ao fazer a criação de um novo evento, no campo da data do evento poderia surgir um calendário para o utilizador selecionar a data pretendida, e no campo de local, poderia ser possível selecionar a localização através do toque num determinado ponto do mapa. Atualmente a *App* PT-UA funciona com uma comunidade global, mas seria seriamente interessante implementar a funcionalidade de esta funcionar em pequenas comunidades mais específicas, sendo que teria acesso a alguns conteúdos específicos de um determinado curso ou cursos dessa comunidade. Ao funcionar como uma comunidade global existe um excesso de informação, sendo que existe informações que apenas interessam a elementos de determinada unidade orgânica, além de não permitir uma adaptação às necessidades específicas de diferentes unidades orgânicas e cursos.

De notar que embora à priori o autor desta dissertação tivesse alguma experiência em *laravel* e *react*, nunca tinha trabalhado com *react native* nem *redux*. Novos conhecimentos foram adquiridos durante este percurso, nomeadamente os que lhe permitiram obter a graduação do *react nanodegree* da *Udacity*. Algumas dificuldades refletiram-se em conseguir implementar o sistema de *likes*, comentários, o *upload* de imagens pelo *react native*, assim como o sistema de navegação *GPS*.

Bibliografia

- Achiiva. (2016). Achiiva. Retrieved February 3, 2018, from <https://www.achiiva.com/>
- Admin Unibly. (2017). Unibly Demo. Retrieved January 2, 2018, from <https://www.youtube.com/watch?v=7KccW0vwDdY>
- Adobe Systems Incorporated. (2017). Introducing Adobe XD. Design. Prototype. Experience. Retrieved December 10, 2017, from www.adobe.com/pt/products/xd.html
- Alarcão, M. (2015). *Insucesso académico e abandono escolar. Seminário Sucesso Académico*. Coimbra.
- Allaboutux.org. (n.d.-a). 3E (Expressing Experiences and Emotions). Retrieved January 3, 2018, from www.allaboutux.org/3e-expressing-experiences-and-emotions
- Allaboutux.org. (n.d.-b). AXE (Anticipated eXperience Evaluation). Retrieved January 3, 2018, from <http://www.allaboutux.org/axe-anticipated-experience-evaluation>
- Allaboutux.org. (n.d.-c). Emocards. Retrieved January 3, 2018, from www.allaboutux.org/emocards
- Allaboutux.org. (n.d.-d). Exploration test. Retrieved January 3, 2018, from www.allaboutux.org/exploration-test
- Allaboutux.org. (n.d.-e). Home. Retrieved January 3, 2018, from <http://www.allaboutux.org>
- Allaboutux.org. (n.d.-f). Self Assessment Manikin (SAM). Retrieved January 3, 2018, from <http://www.allaboutux.org/self-assessment-scale-sam>
- Allaboutux.org. (n.d.-g). User Experience definitions. Retrieved January 3, 2018, from <http://www.allaboutux.org/ux-definitions>
- Android. (n.d.). Componentes da arquitetura do Android. Retrieved December 11, 2017, from <https://developer.android.com/develop/index.html>
- Apache 2. (n.d.-a). Color system. Retrieved December 10, 2017, from <https://material.io/guidelines/style/color.html#color-color-system>
- Apache 2. (n.d.-b). Duration & easing. Retrieved December 10, 2017, from <https://material.io/guidelines/motion/duration-easing.html#>
- Apache 2. (n.d.-c). Environment. Retrieved December 10, 2017, from <https://material.io/guidelines/material-design/environment.html#>
- Apache 2. (n.d.-d). Icons. Retrieved December 10, 2017, from <https://material.io/guidelines/style/icons.html>

- Apache 2. (n.d.-e). Imagery. Retrieved December 10, 2017, from
<https://material.io/guidelines/style/imagery.html>
- Apache 2. (n.d.-f). Implications of motion. Retrieved December 10, 2017, from
<https://material.io/guidelines/motion/material-motion.html#material-motion-implications-of-motion>
- Apache 2. (n.d.-g). Joining & Dividing. Retrieved December 10, 2017, from
<https://material.io/guidelines/motion/transforming-material.html#transforming-material-joining-dividing>
- Apache 2. (n.d.-h). Movement. Retrieved December 10, 2017, from
<https://material.io/guidelines/motion/movement.html>
- Apache 2. (n.d.-i). Movement of material. Retrieved December 10, 2017, from
<https://material.io/guidelines/material-design/material-properties.html#material-properties-movement-of-material>
- Apache 2. (n.d.-j). Object relationships. Retrieved December 10, 2017, from
<https://material.io/guidelines/material-design/elevation-shadows.html#elevation-shadows-object-relationships>
- Apache 2. (n.d.-k). Principles. Retrieved December 10, 2017, from
<https://material.io/guidelines/#introduction-principles>
- Apache 2. (n.d.-l). Typography. Retrieved December 10, 2017, from
<https://material.io/guidelines/style/typography.html>
- Apache Cassandra. (2016). Apache Cassandra. Retrieved January 2, 2018, from
<http://cassandra.apache.org/>
- Apple Inc. (2010). Why Objective-C? Retrieved December 10, 2017, from
https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/OOP_ObjC/Articles/ooWhy.html#//apple_ref/doc/uid/TP40005149-CH3-SW1
- Apple Inc. (2017a). About Swift. Retrieved December 10, 2017, from
https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/index.html#//apple_ref/doc/uid/TP40014097-CH3-ID0
- Apple Inc. (2017b). Action Sheets. Retrieved December 10, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/views/action-sheets/>
- Apple Inc. (2017c). Adaptivity and Layout. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/visual-design/adaptivity-and-layout/>

Apple Inc. (2017d). Alerts. Retrieved December 10, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/views/alerts/>

Apple Inc. (2017e). Animation. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/visual-design/animation/>

Apple Inc. (2017f). App Icon. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/icons-and-images/app-icon/>

Apple Inc. (2017g). Branding. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/visual-design/branding/>

Apple Inc. (2017h). Buttons. Retrieved December 10, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/controls/buttons/>

Apple Inc. (2017i). Color. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/visual-design/color/>

Apple Inc. (2017j). Labels. Retrieved December 10, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/controls/labels/>

Apple Inc. (2017k). Launch Screen. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/icons-and-images/launch-screen/>

Apple Inc. (2017l). Maps. Retrieved December 10, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/views/maps/>

Apple Inc. (2017m). Navigation Bars. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/bars/navigation-bars/>

Apple Inc. (2017n). Scroll Views. Retrieved December 10, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/views/scroll-views/>

Apple Inc. (2017o). Search Bars. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/bars/search-bars/>

Apple Inc. (2017p). Sliders. Retrieved December 10, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/controls/sliders/>

Apple Inc. (2017q). Tab Bars. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/bars/tab-bars/>

Apple Inc. (2017r). Tables. Retrieved December 10, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/views/tables/>

Apple Inc. (2017s). Terminology. Retrieved December 9, 2017, from
<https://developer.apple.com/ios/human-interface-guidelines/visual-design/terminology/>

- Apple Inc. (2017t). Text Views. Retrieved December 10, 2017, from <https://developer.apple.com/ios/human-interface-guidelines/views/text-views/>
- Apple Inc. (2017u). Toolbars. Retrieved December 10, 2017, from <https://developer.apple.com/ios/human-interface-guidelines/bars/toolbars/>
- Apple Inc. (2017v). Typography. Retrieved December 9, 2017, from <https://developer.apple.com/ios/human-interface-guidelines/visual-design/typography/>
- Apple Inc. (2017w). UI Design Do's and Don'ts. Retrieved December 9, 2017, from <https://developer.apple.com/design/tips/>
- Apple Inc. (2017x). Web Views. Retrieved December 10, 2017, from <https://developer.apple.com/ios/human-interface-guidelines/views/web-views/>
- Arah, T. (2006). CMS: content management systems. Retrieved December 12, 2017, from http://designer-info.com/Web/cms_introduction.htm
- Arda. (2014). bash - CentOS colors in php-cli applications - Server Fault. Retrieved December 12, 2017, from <https://serverfault.com/questions/591105/centos-colors-in-php-cli-applications>
- ATTRACT Project. (2012). *ATTRACT - Enhancing the attractiveness of studies in science and technology*. Sweden. Retrieved from <http://www.attractproject.org/>
- Axure Software Solutions Inc. (2017). Design the right solution. Retrieved December 10, 2017, from <https://www.axure.com/>
- Balsamiq Studios, L. (2017). Balsamiq. Retrieved December 10, 2017, from <https://balsamiq.com/>
- Bos, B. (2016). A brief history of CSS until 2016. Retrieved December 9, 2017, from <https://www.w3.org/Style/CSS20/history.html>
- Briones, R. L., & Janoske, M. (2013). Mentoring 2.0: how PR educators use social media to create and maintain relationships with students. *International Journal of Continuing Engineering Education and Life-Long Learning*, 23(1), 18–32.
<http://doi.org/10.1504/IJCEELL.2013.051764>
- Bridgepoint Education (2017). Ashford University Mobile on the App Store. Retrieved December 13, 2017, from <https://itunes.apple.com/us/app/ashford-university-mobile/id440722504?mt=8>
- Carvalhosa, S. (n.d.). Promoção do sucesso académico no ISCTE-IUL. Lisboa.
- Chronus. (2018). Chronus. Retrieved February 3, 2018, from <https://chronus.com/>
- CodeIgniter. (n.d.). CodeIgniter. Retrieved January 2, 2018, from <https://codeigniter.com/>
- Connolly, D. (2004). Hypertext Transfer Protocol -- HTTP/1.1. Retrieved January 2, 2018, from <https://www.w3.org/Protocols/rfc2616/rfc2616.html>

- Coutinho, C. (2006). Aspectos metodológicos da investigação em tecnologia educativa em Portugal (1985-2000). *Actas Do XIV Colóquio AFIRSE*, 1--12. Retrieved from <http://repositorium.sdum.uminho.pt/handle/1822/6497>
- Coutinho, C., & Chaves, J. (2001). Desafios à investigação em TIC na educação: as metodologias de desenvolvimento. *II Conferência Internacional Challenges'2001 Desafios'2001*, 895–903. Retrieved from <http://hdl.handle.net/1822/4277>
- Cullen, K. (2012). On Typefaces. In *Design Elements -Typography Fundamentals* (pp. 52–81). Beverly: Rockport Publishers, Inc.
- Dong, Z. (2011). Aspect oriented programming technology and the strategy of its implementation. *2011 International Conference on Intelligence Science and Information Engineering*, 457–460. <http://doi.org/10.1109/ISIE.2011.44>
- Edmodo. (2018). About Edmodo. Retrieved January 2, 2018, from <https://www.edmodo.com/about>
- Education, B. (n.d.). Ashford University Mobile.
- Ejsmont, A. (2015a). Core concepts. In B. Wilder, T. Wilder, & D. D. Coward (Eds.), *Web scalability for startup engineers - Tips & techniques for scalling your web application* (pp. 1–35). Allite Books.
- Ejsmont, A. (2015b). Principles of good software design. In B. Wilder, T. Wilder, & D. D. Coward (Eds.), *Web scalability for startup engineers - Tips & techniques for scalling your web application* (pp. 37–81). Allite Books.
- eMentorConnect. (n.d.). eMentorConnect. Retrieved February 3, 2018, from <http://ementorconnect.com/our-company/>
- Fernandes, J. M. (2013). *Programa de Tutoria da Universidade de Aveiro*. Universidade de Aveiro.
- Graduway. (2017). Graduway. Retrieved February 4, 2018, from <https://graduway.com/>
- Griol, D., Molina, J. M., & Callejas, Z. (2017). Incorporating android conversational agents in m-learning apps. *Expert Systems*, 34(4), 1–17. <http://doi.org/10.1111/exsy.12156>
- Guide Book Edu. (n.d.). GuideBook Edu. Retrieved January 2, 2018, from <https://guidebook.com/schools/>
- Haber-Curran, P., Everman, D., & Martinez, M. (2017). Mentors' personal growth and development in a college access mentorship program. *Mentoring & Tutoring: Partnership in Learning*, 25(4), 486–503. <http://doi.org/10.1080/13611267.2017.1403558>
- Hall, W., & Tiropanis, T. (2012). Web evolution and web science. *Computer Networks*, 56(18), 3859–3865. <http://doi.org/10.1016/j.comnet.2012.10.004>

- Hou, B., Qian, K., Li, L., Shi, Y., Tao, L., & Liu, J. (2016). MongoDB NoSQL injection analysis and detection. *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing*, 75–78. <http://doi.org/10.1109/CSCloud.2016.57>
- I eat css & Vandelay Web (2015). Zurb Foundation Framework Tutorial: Step-by-Step Tutorial. Retrieved December 13, 2017, from <http://ieatcss.com/zurb-foundation-framework.html>
- Instituto Politécnico de Setúbal. (2015). *A promoção do sucesso escolar no Instituto Politécnico de Setúbal*. Setúbal.
- InVision. (2017). InVision. Retrieved December 10, 2017, from www.invisionapp.com
- Johansson, M. P. (2016). var, let and const — What, why and how — ES6 JavaScript Features. Retrieved December 9, 2017, from <https://medium.com/humans-create-software/var-let-and-const-what-why-and-how-es6-javascript-features-92f1833f56d0>
- Jqueryhouse. (n.d.). JavaScript Libraries for Creating Beautiful Animations | jQueryHouse. Retrieved December 14, 2017, from <https://jqueryhouse.com/javascript-animation-libraries/>
- Justinmind. (2017). All-in-one Prototyping Tool for web and mobile apps. Retrieved December 10, 2017, from www.justinmind.com
- Kadry, S., & Roufayel, R. (2017). How to use effectively smartphone in the classroom. *IEEE Global Engineering Education Conference, EDUCON*, 441–447. <http://doi.org/10.1109/EDUCON.2017.7942884>
- Kumar, A., Kumar, A., & Iyyappan, M. (2016). Applying separation of concern for developing softwares using aspect oriented programming concepts. *Procedia Computer Science*, 85, 906–914. <http://doi.org/10.1016/j.procs.2016.05.281>
- Laravel. (n.d.). Laravel. Retrieved January 2, 2018, from www.laravel.com
- Llauradó, O. (2015). Escala de Likert: O qué é e como utilizá-la. Retrieved from <https://www.netquest.com/blog/br/escala-likert>
- McGinnis, T. (n.d.). The History of JavaScript: ECMAScript, TC39, and beyond. Retrieved December 9, 2017, from <https://medium.freecodecamp.org/ecmascript-tc39-and-the-history-of-javascript-26067498feb9>
- Mentor Scout. (2018). We help companies take their mentoring programs to new heights. Retrieved February 4, 2018, from <https://www.mentorscout.com/>
- Mentored. (2016). Mentored. Retrieved January 2, 2018, from www.mentored.com
- Mentored. (2017). mentored. Retrieved January 2, 2018, from <https://mentored.com/platform.html>
- MentorNet. (n.d.). MentorNet Program Levels. Retrieved February 4, 2018, from

- <http://mentornet.org/organization/programs.html>
- Merritt, S. M., & Havill, L. (2016). Electronic and face-to-face communication in mentoring relationships. *Development and Learning in Organizations: An International Journal*, 30(3), 17–20. <http://doi.org/10.1108/DLO-10-2015-0088>
- MongoDB. (2018). What is MongoDB? Retrieved January 2, 2018, from <https://www.mongodb.com/what-is-mongodb>
- Moreira, G. (n.d.). *Projeto FICA e o sucesso académico*. Aveiro, Portugal.
- Mr. Miranda The Tech Guru. (2017). Edmodo App Tutorial 2017. Retrieved January 2, 2018, from <https://www.youtube.com/watch?v=WDxX5BD6ntY>
- Nielsen, J. (1995). 10 Usability Heuristics for User Interface Design. Retrieved January 3, 2018, from <https://www.nngroup.com/articles/ten-usability-heuristics/>
- Nielsen, J. (2012). Usability 101: Introduction to Usability. Retrieved January 3, 2018, from www.nngroup.com/articles/usability-101-introduction-to-usability
- Norman, D., & Nielsen, J. (2018). The Definition of User Experience (UX). Retrieved January 3, 2018, from www.nngroup.com/articles/definition-user-experience
- Northern Illinois University. (2011). NIU Mobile App - Northern Illinois University. Retrieved January 2, 2018, from <https://www.youtube.com/watch?v=z65icSpfPiY>
- Oracle. (n.d.). Oracle. Retrieved January 2, 2018, from <https://www.oracle.com/br/mysql/index.html>
- Pimmer, C., Chipps, J., Brysiewicz, P., Walters, F., Linxen, S., & Gröhbriel, U. (2017). Facebook for supervision? Research education shaped by the structural properties of a social media space. *Technology, Pedagogy and Education*, 26(5), 517–528. <http://doi.org/10.1080/1475939X.2016.1262788>
- Pływaczewski, E. W., & Kraśnicka, I. (2016). Legal education in transition: Is the bologna process responding to Europe's place in the world? In *Women and Children as Victims and Offenders: Background, Prevention, Reintegration* (pp. 335–361). Scopus. http://doi.org/10.1007/978-3-319-08398-8_13
- Programa de Tutoria / PT-UA. (2018). Programa de Tutoria / PT-UA - Página inicial. Retrieved from <https://www.facebook.com/tutoria.ua/photos/pb.261739790626700.-2207520000.1539214937.1379637988836869/?type=3&theater>
- Queiroz, J. (2015). *Que propostas para a promoção do sucesso académico no 1º. Ciclo de estudos?* Lisboa.

- Quora. (2016). What are the pros and cons of using JavaScript ES5 versus ES6? Retrieved January 2, 2018, from <https://www.quora.com/What-are-the-pros-and-cons-of-using-JavaScript-ES5-versus-ES6>
- Rivera, R. (2010). Instruction over online social networks: Where does the platform lead? 2010 *International Conference on Technology for Education, T4E 2010*, 209–215. <http://doi.org/10.1109/T4E.2010.5550096>
- Saleh, H., Holmes, E., Bray, T., & Yusuf, S. (2016a). An introduction to Apache Cordova. In *Mobile application development: Javascript frameworks* (pp. 3–24). Birmingham: Packt Publishing Ltd. Retrieved from <http://www.soel.ru/cms/f/325054.rar>
- Saleh, H., Holmes, E., Bray, T., & Yusuf, S. (2016b). Exploring the sample application. In *Mobile application development: Javascript frameworks* (pp. 299–310). Birmingham: Packt Publishing Ltd. Retrieved from <http://www.soel.ru/cms/f/325054.rar>
- Saleh, H., Holmes, E., Bray, T., & Yusuf, S. (2016c). Understanding react native fundamentals. In *Mobile application development: Javascript frameworks* (pp. 311–318). Birmingham: Packt Publishing Ltd. Retrieved from <http://www.soel.ru/cms/f/325054.rar>
- Santos, C. (2016). *Web APIs*. Aveiro, Portugal. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/API>
- Santos, C. (2017). *Avila Crew - Uma experiência de tutoria de estudantes para estudantes*. Aveiro, Portugal.
- Shappir, D. (2016). Dan Shappir: Things you can do in ES6 that can't be done in ES5. Retrieved January 2, 2018, from <https://pt.slideshare.net/DanielleAVincent/dan-shappir-things-you-can-do-in-es6-that-cant-be-done-in-es5>
- Simões, D., Pinheiro, M. M., & Moreira, G. (2016). Experiences in higher education – lessons learned from a tutorial program. In L. G. Chova, A. L. Martínez, & I. C. Torres (Eds.), *INTED2016, Proceedings of the 10th International Technology, Education and Development Conference* (pp. 5644–5652). Valencia: IATED Academy. <http://doi.org/10.21125/inted.2016>
- Smilansky, O. (2016). Create an addictive mobile application. *Costumer Relationship Management*, 26–29.
- Some Character LLC. (n.d.). Mockingbird: get your ideas out of your head. Retrieved December 10, 2017, from <https://gomockingbird.com/>
- SQLite. (n.d.). SQLite. Retrieved January 2, 2018, from <https://www.sqlite.org/features.html>
- Stone, T. L., Adams, S., & Morioka, M. (2006). Color Meanings. In *Color Design Workbook: a real-world guide to using color in graphic design* (pp. 24–31). Massachusetts: Rockport

- Publishers, Inc.
- Symeonidis, P., & Tiakas, E. (2014). Transitive node similarity: Predicting and recommending links in signed social networks. *World Wide Web*, 17(4), 743–776.
<http://doi.org/10.1007/s11280-013-0228-2>
- Tech Terms. (2012). App. Retrieved January 2, 2018, from <https://techterms.com/definition/app>
- Teixeira, F. (2017). Muito além do teste de usabilidade: os vários tipos de pesquisas com usuários em UX. Retrieved February 4, 2018, from <https://brasil.uxdesign.cc/muito-além-do-teste-de-usabilidade-os-vários-tipos-de-pesquisas-com-usuários-em-ux-b91a6e15bc61>
- The PHP Group. (2018). O que é o PHP? Retrieved January 2, 2018, from https://secure.php.net/manual/pt_BR/intro-what-is.php
- Unibly. (n.d.). Mentoring Platform — Unibly. Retrieved January 2, 2018, from <http://unibly.squarespace.com/mentoringplatform>
- Unibly. (2015). Unibly. Retrieved January 2, 2018, from <http://www.unibly.com/>
- Universidade de Aveiro. (2016). Sessão no âmbito do ciclo Pedagogias Ativas @ UA e do projeto FICA - Explorando o conceito de ambiente pessoal de aprendizagem no ensino superior. Retrieved January 1, 2018, from <https://uaonline.ua.pt/pub/detail.asp?c=46058>
- Universidade de Coimbra. (2010). Implementação do Processo de Bolonha na UC. Retrieved January 1, 2018, from <http://www.uc.pt/ge3s/destaques/bolonhauc2010>
- User Experience Professionals Association. (n.d.). About UX. Retrieved January 3, 2017, from www.uxpa.org/resources/about-ux
- W3.org. (2017). History. Retrieved January 2, 2018, from https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript
- W3C. (n.d.). What is HTML? Retrieved December 8, 2017, from <https://www.w3.org/html/>
- Weerasinghe, P., Marasinghe, A., Ranaweera, R., Amarakeerthi, S., & Cohen, M. (2013). Emotion expression for affective social communication. *Proceedings - 2013 International Conference on Biometrics and Kansei Engineering, ICBAKE 2013*, 148–153.
<http://doi.org/10.1109/ICBAKE.2013.31>
- Wong, E. (2017). Shneiderman's eight golden rules will help you design better interfaces. Retrieved January 3, 2018, from <https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces>

Apêndice A: Disponibilização do código

Para aceder aos códigos, é necessário visitar os endereços *Web* e pedir autorização para conseguir visualizar e descarregar.

No endereço https://github.com/nunosoares27/PT-UA_React-Native está disponível o código da versão *App*, sendo necessário após o download entrar dentro da pasta a partir da linha de comandos e correr o comando `npm install` e posteriormente `npm start`. Seguir atentamente as indicações do terminal para especificar as opções.

No endereço https://github.com/nunosoares27/PT-UA_Admin está disponível o código da versão *Web*. Esse código não dá para executar, uma vez que não foram incluídos no repositório as referências a dados sensíveis como os dados de acesso ao servidor e sua base de dados.